1 LÓGICAS AUXILIARES

Los métodos formales son usados en las Ciencias de la Computación para verificar que un programa sea correcto y funcione conforme a nuestras expectativas. Desde esta perspectiva, el objetivo de la **lógica** es desarrollar lenguajes para modelar las situaciones que nos encontramos como profesionales de la computación. Razonar acerca de estas situaciones significa construir argumentos acerca de ellas, argumentos formales que puedan ser defendidos rigurosamente o ser ejecutados por una computadora [107]. Dentro de esta tradición, encontramos dos enfoques, con propósitos ligeramente distintos[1], para **especificar** formalmente a los agentes racionales:

Lógica y Computación

Especificación de agentes

- **Lenguajes orientados a agentes.** En este enfoque, el lenguaje empleado para especificar formalmente a los agentes y su comportamiento, es a su vez, un lenguaje que el agente puede ejecutar. Por ejemplo, Agento [168, 167], Golog y Congolog [116, 75], AgentSpeak(L) [152], 3APL [98] y más recientemente 2APL [46].
- **Lenguajes externos a los agentes.** En este enfoque, los métodos formales conducen a metalenguajes usados para especificar el diseño de los agentes y para verificar que ciertas propiedades del comportamiento de los agentes se cumplan. Por ejemplo, las diferentes **lógicas BDI** [40, 149, 151, 154, 94, 176], incluyendo Lora [185]; o las diferentes especificaciones de agentes en lenguaje Z [50, 51, 53].

Lógicas BDI

Lo ideal sería contar con formalismos que atendieran ambos propósitos, pero el compromiso entre **expresividad y costo computacional**, hacen que esto sea en extremo difícil [175]. Debido a las propiedades que esperamos de un agente, en particular reactividad, los lenguajes orientados a agentes deben ser computacionalmente eficientes; pero en contra parte, otras propiedades de los agentes como la iniciativa y la habilidad social, requieren de lenguajes muy expresivos para obtener sus especificaciones formales.

Expresividad vs costo computacional

La expresividad demandada a las lógicas para especificar a los agentes racionales como sistemas Intencionales, provoca que éstas suelan tener diversos **componentes**, cada uno de ellos un método formal particular. Los componentes incluyen:

Métodos formales

- Un componente **proposicional** o de **primer orden** [70], para tratar con el contenido de las actitudes del agente.
- Un componente **modal** [104], para tratar con las actitudes del agente: Creencias, deseos e intenciones.
- un componente **dinámico** [95], para abordar la ejecución de las acciones y la ocurrencia de eventos;

1 Para una discusión más detallada al respecto y la consideración de algunos casos de estudio, vean el capítulo 9 de libro de Wooldridge [185].

- Un componente **temporal**, basado generalmente en las lógicas temporales ramificadas CTL y $CTL*$ [55], para tratar los cambios de los componentes anteriores en el tiempo.

Tal aparato es lo suficientemente expresivo, como para definir formalmente los conceptos de la teoría de razonamiento práctico de Bratman [22] y, desgraciadamente, lo suficientemente complejo como para que un agente razone directamente haciendo uso de él. Se recomienda la revisión de las técnicas conocidas como de refinamiento, por ejemplo, la especificación en lenguaje Z [118] que d'Inverno y col. [50] hacen de dMARS.

Este capítulo introduce formalmente, aunque de manera muy breve, los componentes no Intencionales de las llamadas lógicas BDI. Para ello tres aspectos son considerados:

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

  Alfabeto
  Sintaxis
  fbf

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

  Axiomas
  Reglas de inferencia
  Teoremas

- La **teoría del modelo** establece la interpretación de las fbf en un sistema formal. Su función es relacionar las fbf con alguna representación simplificada de la realidad que nos interesa, estableciendo su valor de verdad. Esta versión de realidad corresponde a lo que informalmente llamamos "modelo", sin embargo, en lógica el significado de "modelo" está íntimamente relacionado con el lenguaje del sistema formal: si una interpretación M hace que la expresión α sea verdadera, se dice que que M es un **modelo** de α o que M satisface α, y se escribe $M \models \alpha$. Se dice que una fbf α es **válida**, si y sólo si, toda interpretación M es un modelo de ella. Lo anterior se denota por $M \vdash \alpha$. El símbolo α se usa aquí como una variable **meta lógica**, es decir una variable que tiene como referente el lenguaje del sistema formal mismo, y por lo tanto, no forma parte de el. En lo que sigue se usaran letras griegas como variables meta lógicas.

  Modelo
  Validez
  Variables meta lógicas

Los elementos del lenguaje y la teoría de la prueba, constituyen la sintaxis del sistema formal y su **base axiomática**. La teoría del modelo está relacio-

Base axiomática

nada con la **semántica** del sistema formal. *Semántica*

El orden de exposición es el siguiente: Haremos un recordatorio rápido al cálculo proposicional y al de predicados, la lógica modal. Con esta base, se abordarán posteriormente las lógicas temporales lineales y arborescentes.

1.1 LÓGICA PROPOSICIONAL

La **Lógica Proposicional** (LP) es utilizada para representar información declarativa. Las fórmulas de este lenguaje se construyen a partir de enunciados o **proposiciones** atómicas que expresan hechos acerca del mundo, es decir, información efectiva que puede ser declara como falsa o verdadera; y algunos **operadores** sobre ellas, por ejemplo: disyunción ($\vee$), conjunción ($\wedge$), negación ($\neg$), implicación material ($\rightarrow$), etc. A continuación, la sintaxis y la semántica del cálculo proposicional son detallados:

Proposiciones atómicas

Operadores lógicos

1.1.1 Sintaxis

Sintaxis LP 1. *El lenguaje de la lógica proposicional $\mathcal{L}_{LP}$, se compone a partir del alfabeto $\{p, q, r, \dots\} \cup \{p_1, p_2, p_3, \dots\} \cup \{\top, \bot, \neg, \wedge, \vee, \rightarrow, \equiv, (,)\}$ y la siguiente gramática en forma Backus Naur (BNF):*

$$\phi ::= p \mid (\neg\phi) \mid (\phi \vee \phi)$$

donde p denota una variable proposicional y ϕ denota fórmulas bien formadas. Es decir, cada ocurrencia de ϕ a la derecha de $::=$ denota una fórmula previamente construída. $Var = \{p, q, r, \dots\} \cup \{p_1, p_2, p_3, \dots\}$ denota el conjunto infinito de **variables proposicionales**. El resto de los operadores pueden definirse como de costumbre:

Variables proposicionales

Definición 1.1 (conjunción). $(\alpha \wedge \beta) =_{def} \neg(\neg\alpha \vee \neg\beta)$;

Definición 1.2 (implicación material). $(\alpha \rightarrow \beta) =_{def} (\neg\alpha \vee \beta)$;

Definición 1.3 (equivalencia material). $(\alpha \equiv \beta) =_{def} ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$;

Definición 1.4 (falso). $\bot =_{def} \neg\alpha \wedge \alpha$;

Definición 1.5 (verdadero). $\top =_{def} \neg\bot$

Ejemplo 1.1. *Las siguientes expresiones son fbf de la lógica proposicional:*

- p
- $\neg p \wedge q$
- $(p \wedge q) \rightarrow (q \vee r)$

Las siguientes expresiones no son fbf de la lógica proposicional:

- $\neg \vee p$
- $\rightarrow q \neg r$
- $\vee a \wedge b$

1.1.2 Semántica

Valor de verdad

Las proposiciones tienen asociado un **valor de verdad**, de forma que toda proposición es verdadera o su negación es verdadera, lo que se conoce como el **principio del tercero excluído** (*tertium non datur*). Por lo tanto, una expresión que no es falsa, ni verdadera, o puede ser verdadera y falsa, no es una proposición.

Principio del tercero excluido

Ejemplo 1.2. *Las siguientes expresiones son ejemplos de proposiciones:*

p : *Gané la lotería la semana pasada.*

q : *Compré un billete de lotería.*

r : *Gané en las carreras de caballos.*

Las siguientes no lo son:

s : *Cierra la puerta.*

t : *¿Qué hora es?*

u : *!Cojones!*

Los operadores pueden usarse para formar proposiciones compuestas. Aplicando el operador de negación a la proposición p produce una nueva proposición que describe el hecho de que no compré un billete de lotería: $\neg p$. Observen que el valor de verdad de esta expresión depende exclusivamente del valor de verdad de la proposición p. Esto aplica a todas las proposiciones compuestas creadas con conectivos: sus valores de verdad dependen de sus argumentos. Por ello, se dice que los conectivos de la lógica proposicional son **funcionales** con respecto al valor de verdad (*truth-functionals*) –El valor de verdad de las proposiciones compuestas depende exclusivamente de sus operadores y sus argumentos.

Operadores funcionales

La negación ($\neg$), la disyunción ($\vee$) y la conjunción ($\wedge$), cuando se aplican de esta manera, son conocidos como **conectivos** y las proposiciones sobre las cuales operan se conocen como sus **argumentos**. Los operadores que sólo requieren un argumento se conocen como **unarios**, ej. la negación. Los operadores que requieren dos argumentos se conocen como **binarios**, ej. disyunción y conjunción.

La interpretación de los otros conectivos es como sigue. Dados p y r, $p \vee r$ denota que al menos una de las dos proposiciones es verdadera; es decir, la **disyunción** de r y p: Gané la lotería la semana pasada ó gané en las carreras de caballos. la fórmula $p \wedge r$, dual a la anterior, denota la **conjunción** de p y r: Gané la lotería la semana pasada y gané en las carreras de caballos. El enunciado Si gané a la lotería la semana pasada entonces compré un billete de lotería, expresa una **implicación** entre p y q, sugiriendo que q es una consecuencia lógica de p. Esto se denota por $p \rightarrow q$, donde p se conoce como **antecedente** y q como **consecuente** de la expresión. Por supuesto, que se pueden construir expresiones anidadas como:

$$p \wedge q \rightarrow \neg r \vee q$$

que significa: Si gané la lotería la semana pasada y compré un billete de lotería, entonces no gané en las carreras de caballos ó compré un billete

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

Valores de verdad

Definición 1.6 (Modelo y Valores de verdad). *El conjunto de **valores de verdad** contiene dos elementos $\top$ and $\bot$, donde $\top$ denota verdadero y $\bot$ falso. Abusando un poco del término, el **modelo** de una fórmula ϕ es una asignación de un valor de verdad a cada proposición en ϕ.*

Modelo

Un método para asignar valores de verdad a las fbfs de la lógica proposicional es hacer uso de **tablas de verdad** (Cuadro 1.1).

Tablas de verdad

ϕ	ψ	$\phi \wedge \psi$	$\phi \vee \psi$	$\phi \rightarrow \psi$	$\neg\phi$	$\top$	$\bot$
T	T	T	T	T	F	T	F
T	F	F	T	F			
F	T	F	T	T			
F	F	F	F	T			

Cuadro 1.1: Tablas de verdad para los operadores de la Lógica Proposicional

Para una fbf ϕ que contiene las proposiciones $p_1, p_2, \ldots, p_n$, podemos construir en principio su tabla de verdad, sólo que serán necesarios 2^n renglones para ello. Cada renglón enumera una interpretación de la fbf en cuestión.

Número de interpretaciones

Sea $M_{LP} =_{def} \langle L \rangle$ el modelo para la Lógica Proposicional (nuevamente, abusando un poco del término). $L \subseteq Var$ es en realidad una **interpretación** o etiquetado (*label*), que identifica el conjunto de variables proposicionales que tienen valor verdadero. Luego, la semántica de las fbf del cálculo proposicional es como sigue:

Interpretación

Semántica LP 1. $M_{LP} \models \alpha$ *Si y sólo si* $\alpha \in L$, *donde* $\alpha \in Var$

Semántica LP 2. $M_{LP} \models \neg\alpha$ *Si y sólo si* $M_{LP} \not\models \alpha$

Semántica LP 3. $M_{LP} \models (\alpha \vee \beta)$ *Si y sólo si* $M_{LP} \models \alpha$ *ó* $M_{LP} \models \beta$

Decimos que M_{LP} es un modelo, porque la interpretación L hace que la fórmula en cuestión sea verdadera (en otro caso, M_{LP} no sería su modelo).

Ejemplo 1.3. *La fbf $p \vee q$ es verdadera en la interpretación $L = \{p\}$. De hecho las interpretaciones donde p ó q pertenecen a L son modelos de esta fbf.*

Si una fbf α es verdadera en toda interpretación posible decimos que la fórmula es **válida**, lo cual se denota $\vdash \alpha$. Esto quiere decir que todas las interpretaciones (todos los renglones de la tabla de verdad de la expresión) son modelos de la fórmula en cuestión.

Validez

Ejemplo 1.4. *La siguiente fbf válida, expresa el principio del tercero excluído a la manera de Russell y Whitehead en* Principia Mathematica*:*

$$\vdash p \vee \neg p$$

De hecho puede substituirse por el patrón:

$$\vdash \alpha \vee \neg \alpha$$

Por cuestiones de espacio y tiempo, no abordaremos la teoría de prueba de la lógica proposicional aquí. El lector interesado puede consultar las notas del curso de Representación del Conocimiento para ello [2]. Solo mencionaré que el *Modus Ponens* es, entre otras, una regla de inferencia de este sistema:

Modus Ponens

$$\frac{\alpha \quad \alpha \rightarrow \beta}{\beta}(\mathbf{MP})$$

Resulta curioso que el sistema resultante sea omnisciente, en el sentido de que deriva por (MP) las consecuencias de todas las **tautologías** (fbf válidas).

Tautología

1.2 LÓGICA DE PRIMER ORDEN

Básicamente, la lógica de primer orden, extiende la lógica proposicional al introducir símbolos que nos permiten expresarnos acerca de los **objetos** en un dominio de discurso dado y las **relaciones** entre ellos. El conjunto de todos estos objetos se conoce como **universo de discurso** y se denota como $\mathcal{U}$. Los miembros del universo de discurso pueden ser objetos concretos, ej., un libro, un robot, etc; abstractos, ej., números; e incluso, ficticios, ej., unicornios, etc. Un objeto es algo sobre lo cual queremos expresarnos. Para ejemplificar esto, consideren el multi citado mundo de los bloques [70] que se muestra en la figura 1.1. El universo de discurso para esta conceptualización de tal escenario es el conjunto que incluye los cinco bloques y la mesa: $\{a, b, c, d, e, mesa\}$.

Objetos
Relaciones
Universo de discurso

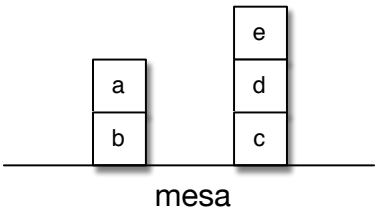

Figura 1.1: El mundo de los bloques.

Definición 1.7 (Función y Base funcional)**.** *Una* ***función*** $F : \mathcal{U}^n \mapsto \mathcal{U}$*, es una manera de denotar objetos en el universo de discurso en términos de su relación a otros objetos. El número n de argumentos de una función se conoce como su* ***aridad****. F/n expresa que la función F tiene n argumentos. Las funciones pueden definirse por* ***extensión****, como el conjunto de todas las tuplas* $n+1$ *que satisfacen la*

Función
Aridad
Definiciones por extensión e intensión

[2] https://www.uv.mx/personal/aguerra/rc/ Ver los capítulos 3 y 4: Deducción natural y Lógica Proposicional.

*relación; o por **intensión**, en términos de las propiedades que satisfacen la relación. El conjunto de todas las funciones consideradas en la conceptualización del mundo se conoce como **base funcional**.*

Base funcional

Ejemplo 1.5. *La función parcial sombrero/1 regresa el bloque que está inmediatamente encima del bloque que es su argumento. Su extensión es $\{(b,a),(c,d),(d,e)\}$. La función sandwich/2 regresa el bloque que está en medio de sus dos argumentos. Su extensión es $\{(c,e,d),(e,c,d)\}$. La función sucesor/1 para los números naturales se define por intensión: $sucesor(X) \mapsto X+1$ para $X \in \mathbb{N}$; su extensión es infinita. Si estos fuesen todas las funciones definidas en nuestro sistema, su base funcional sería $\{sombrero/1, sandwich/2, sucesor/1\}$.*

Definición 1.8 (Predicado y Base relaciona)**.** *Un **predicado** P/n define una relación entre los n objetos que tiene como argumento. Al igual que en el caso de las funciones, n se conoce como la aridad del predicado. Los predicados pueden verse como funciones booleanas que son verdaderas, si y sólo si, sus argumentos pertenecen a la definición del mismo. El conjunto de todos los predicados usados en la conceptualización se conoce como **base relacional**.*

Predicado

Base relacional

Ejemplo 1.6. *El predicado sobre/2 se satisface para dos bloques, si y sólo si el bloque de su primer argumento está sobre el que es su segundo argumento. Para la escena mostrada en la Figura 1.1, se define por los pares $\{(a,b),(d,c),(e,d)\}$. El predicado libre/1 se satisface para un bloque, si y sólo si éste no tiene ningún bloque encima: $\{a,e\}$. La relación par/1 sobre los naturales de define por intensión $par(X)$ si y sólo si $Xmod2 = 0$.*

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

Además de las funciones y predicados, la flexibilidad el cálculo de predicados viene del uso de cuantificadores y variables. Las **variables**, cuyos valores son objetos del universo de discurso se representan por cualquier secuencia de caracteres que inicie con una mayúscula.

Variables

Ejemplo 1.7. *$X, X1, X', Var, Var1, VarPrivada, \ldots$ denotan variables.*

El **cuantificador universal** ($\forall$), nos permite expresar hechos acerca de todos los objetos en el universo del discurso, sin necesidad de enumerarlos.

Cuantificador universal

Ejemplo 1.8. *$\forall X \forall Y\ sobre(X,Y) \rightarrow porArriba(X,Y)$ expresa que todo objeto sobre otro, está por encima de éste último.*

El **cuantificador existencial** ($\exists$) nos permite expresar la existencia de un objeto en el universo de discurso, con cierta propiedad en particular.

Cuantificador existencial

Ejemplo 1.9. *$\exists X\ libre(X) \wedge sobre(X, mesa)$ expresa que hay al menos un objeto que no tiene bloques sobre él y está en la mesa.*

1.2.1 Sintaxis

El **alfabeto** de la Lógica de Primer Orden se obtienen al extender la lógica proposicional con un conjunto numerable de símbolos de predicados (*Pred*), funciones (*Func*) y constantes. Se asume un conjunto infinito de variables (*Var*) que toman valores en el universo de discurso. $|R|$ denota la aridad del predicado o función R, es decir, su número de argumentos. Las funciones de aridad o se asumen como constantes; los predicados de aridad o se asumen como variables proposicionales. *Alfabeto*

Los **términos** de nuestro lenguaje se forman de variables, constantes y funciones aplicadas a términos. Formalmente: *Términos*

Definición 1.9 (Términos)**.** *Un término se define por:*

$$t ::= x \mid c \mid f(t, \ldots, t)$$

donde $x \in Var$; $c \in Func$ tal que $|c| = 0$; y $f \in Func$ tal que $|f| > 0$.

Los constructores básicos de los términos son las variables y las constantes (funciones de aridad o). Se pueden formar términos más complejos usando functores de aridad mayor a cero, cuyos argumentos son a su vez términos. Por lo tanto, la noción de término depende de su base funcional $Func$.

Ejemplo 1.10. *$calif(hermano(ana), sma)$ denota la calificación obtenida por el hermano de Alex en el curso de Sistemas Multi-Agentes. Observen que los argumentos $hermano(ana)$ y ana son a su vez términos; al igual que sma.*

Con estos elementos podemos definir la **sintaxis** de la Lógica de Primer Orden (LPO): *Sintaxis*

Sintaxis LPO 1. *El lenguaje de la Lógica de Primer Orden $\mathcal{L}_{LPO}$ se construye a partir de las variables Var, los functores Func y los predicados Pred como sigue:*

$$\phi ::= P(t_1, \ldots, t_n) \mid \neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid (\forall x\ \phi) \mid (\exists x\ \phi)$$

donde $P \in Pred$ es un símbolo de predicado de aridad $n \geq 0$; t_i denota términos sobre Func; y $x \in Var$.

Observen que los argumentos de un predicado son siempre términos. Los predicados de aridad cero se asumen como variables proposicionales. El resto de los operadores lógicos se define como en la sección anterior, por tanto, este lenguaje **subsume** al de la lógica proposicional: $LPO \supset LP$. *LPO subsume LP*

1.2.2 Semántica

Cuando usamos cuantificadores, siempre tenemos en mente un universo de discurso $\mathcal{U}$ ó **dominio**. El dominio puede especificarse en término de conjuntos. *Dominio*

Ejemplo 1.11. *Si el dominio $D = \{a, b, c, d, e, mesa\}$, podemos decir que $B = \{a, b, c, d, e\}$ es el conjunto de bloques en D. La fbf $\exists X\ bloque(X) \wedge libre(X)$ en D es equivalente a la fbf $\exists X\ libre(X)$ en B. Se dice que D y B son interpretaciones de estos predicados. Para un predicado de aridad n su aridad estará en $\mathcal{U}^n$.*

Para obtener un **modelo** para la Lógica de Primer Orden formamos el par $\langle D, V\rangle$, donde D es el **dominio** asumido y V es una función, tal que para cualquier predicado de aridad n se obtienen las n-tuplas que corresponden a la interpretación del predicado. En el ejemplo de la Figura 1.1, consideren el predicado *sobre* de aridad dos. Su interpretación es un subconjunto de $B \times B$. Para la escena mostrada, $V(sobre) = \{(a,b),(e,d),(d,c)\}$. Para una constante, la función V regresa la misma constante, ej. $V(a) = a$. Algunas veces la expresión $V(\alpha)$, donde $\alpha \in Const \cup Pred$, se abrevia α^V y se conoce como una **interpretación**.

Modelo
Dominio
Interpretación

Ejemplo 1.12. *Una posible interpretación V para la escena en cuestión es:*

$$\begin{aligned} a^V &= a \\ b^V &= b \\ c^V &= c \\ d^V &= d \\ e^V &= e \\ sobre^V &= \{(a,b),(e,d),(d,c)\} \\ enLaMesa^V &= \{b,c\} \\ libre^V &= \{a,e\} \\ porEncima^V &= \{(a,b),(e,d),(e,c),(d,c)\} \end{aligned}$$

Todo esto puede especificarse formalmente con la siguiente regla semántica:

Semántica LPO 1. *Una interpretación V, con respecto a un dominio de discurso D es una función que satisface las siguientes propiedades: i) Si $\alpha \in Const$, entonces $V(\alpha) = \alpha$; ii) Si $\alpha \in Pred$ es de aridad n, entonces $V(\alpha) \subseteq D^n$.*

Observen que, tal y como se ha definido, las variables no están incluidas en la interpretación. Interpretar las variables de manera separada a los otros símbolos, es una práctica aceptada. En lo que sigue, se asume que las variables en las fbf están acotadas, es decir, bajo el alcance de un cuantificador. Decimos que μ es una **asignación de variables** basada en el modelo $M_1 = \langle D, V\rangle$, si para todo $\alpha \in Var$, $\mu(\alpha) \in D$. Entonces podemos escribir $M_1 \models V_\mu(\alpha)$ para expresar que α se satisface en el modelo $\langle D, V\rangle$ cuando las variables en α toman valores de acuerdo a la asignación μ.

Asignación de variables

Ejemplo 1.13. *Dados el domino D y las interpretación V asumidos, $M_1 \models libre(X)$ cuando $\mu = \{X/a\}$.*

Luego, para las fbf atómicas tenemos que:

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

Semántica LPO 3. *$M_1 \models \neg\alpha$ Si y sólo si $M_1 \not\models \alpha$.*

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

Para formular las reglas semánticas de las expresiones que contienen cuantificadores, necesitamos el concepto de **asignación alternativa de variables**. La idea es que queremos que $V_\mu(\forall x\alpha)$ sea verdadera, no solo cuando $V_\mu(\alpha)$

Asignación alternativa de variables

es verdadera, sino cuando $V_\rho(\alpha)$ es verdadera para cualquier valor en D regresado por cualquier función de asignación ρ. Observen que las variables libres en α deben conservar el mismo valor en tal asignación. Se dice que la asignación de variables ρ es *x-alternativa* de μ, si y sólo si para toda variable α, excepto posiblemente x, $\rho(\alpha) = \mu(\alpha)$. Entonces:

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

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

Validez

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

Principio de remplazamiento

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

Principio de concordancia

1.3 LÓGICA MODAL PROPOSICIONAL

Lógica modal

La **lógica modal** trabaja con conceptos cuyo valor de verdad no es funcional. Estos conceptos generalmente expresan cierto modo, o manera, en la cual una proposición puede ser verdadera. Uno de estos modos es la expresión necesariamente. Podemos agregar un **operador de necesidad** L a la Lógica Proposicional [3], con la siguiente regla sintáctica: Si α es una fbf, también lo es $L\alpha$. Este operador expresa conceptos como necesariamente, debe ser, está sujeto a, etc. Las **interpretaciones** pueden variar entre un sentido deóntico –Debe ser; Ético –Debería ser el caso que; Epistemológico –Es sabido que; o Temporal –Siempre es el caso que. Lo común en todas estas expresiones es que **no son funcionales** con respecto a su valor de verdad. Es decir, el valor de verdad de la proposición p no es suficiente para determinar el valor de verdad de Lp.

Necesidad

Interpretaciones de L

Carácter no funcional

Posibilidad

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

3 Podríamos agregarlo también a la Lógica de Primer Orden, pero por simplicidad, hablaremos aquí de una Lógica Modal Proposicional.

1.3.1 Sintaxis

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

VARIABLES PROPOSICIONALES: Var;

OPERADORES UNARIOS: $\neg$, L;

OPERADORES BINARIOS: $\vee$;

PARÉNTESIS: (,).

Las reglas de formación son las siguientes:

Sintaxis LMP 1. *Si $\alpha \in Var$ Entonces α es una fbf;*

Sintaxis LMP 2. *Si α es una fbf, entonces $\neg\alpha$ and $L\alpha$ son fbf;*

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

Los demás operadores se definen como de costumbre en la Lógica Proposicional (Ver Pág. 313). El operador de **posibilidad** se define como sigue:

Definición 1.10 (posiblemente)**.** $M =_{def} \neg L \neg \alpha$.

1.3.2 Semántica

La teoría del modelo de las Lógicas Modales está basada en las siguientes tres ideas:

1. Determinar el valor de verdad de una expresión modal implica tomar en consideración estados alternativos a los que realmente se han dado –¿Como podrían haber sido las cosas? ¿Cómo deberían ser? ¿Siempre son? etc;
2. Para cada estado de cosas, existe un conjunto de estados posibles alternativos;
3. En un estado dado $M\alpha$ es verdadero si y sólo si α mismo es verdadero en al menos uno de sus estados posibles, y $L\alpha$ es verdadero si y sólo si α es verdadero en todos sus estados posibles.

Mundos posibles

Accesibilidad

Marco

Los estados posibles se conocen generalmente como **mundos posibles**. Se asume un conjunto no vacío W de ellos. Una relación R de **accesibilidad** entre estos mundos, también es asumida. Se dice que w' es accesible desde w si y sólo si $(w, w') \in R$. La estructura $\langle W, R \rangle$ constituye un **marco**.

Ejemplo 1.14. *La Figura 1.2 muestra un marco donde $W = \{w_0, w_1, w_2\}$ y $R = \{(w_0, w_1), (w_0, w_2)\}$. Las letras en cada mundo posible w_i son las proposiciones que son verdaderas en ese mundo posible.*

Asignación de verdad

Una función de **asignación de verdad** V se define como sigue: $V(\alpha, w) = true$ si y sólo si V asigna verdadero a la proposición α en el mundo w. $V(\alpha, w) = falso$, si y sólo si V asigna falso a la proposición α en el mundo w. Observen que V corresponde a la interpretación L en el cálculo proposicional, pero aquí el valor de verdad de una proposición puede cambiar entre los mundos posibles.

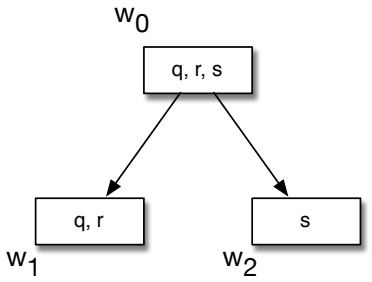

Figura 1.2: Un marco modal conforme a $\langle W, R \rangle$.

Ejemplo 1.15. *En el marco que se muestra en la Figura 1.2, $V(q, w_0) = verdadero$, pero $V(q, w_2) = falso$.*

La función de asignación de verdad también puede definirse también como: $V : W \to 2^{Var}$ denotando el conjunto de variables proposicionales que son verdad en un mundo.

Ejemplo 1.16. *Con respecto al mismo marco: $Var = \{q, r, s\}$; $V(w_0) = \{q, r, s\}$, $V(w_1) = \{q, r\}$, y $V(w_2) = \{s\}$.*

Definición 1.11 (Modelo)**.** *Un **modelo** para la Lógica Modal Proposicional, es una tupla $\langle W, R, V \rangle$ donde $\langle W, R \rangle$ es un marco, y V es una función de asignación de verdad. Tal modelo se conoce como estructura de Kripke, [111]*[4]*.* *Modelo*

Sea $M_2 = \langle W, R, V \rangle$ un modelo, la **semántica** de la LMP se define como sigue: *Semántica*

Semántica LMP 1. $M_2 \models_w \alpha$ *si y sólo si* $\alpha \in V(w)$ *y* $\alpha \in Var$*;*

Semántica LMP 2. $M_2 \models_w \neg\alpha$ *si y solo si* $M_2 \not\models_w \alpha$*;*

Semántica LMP 3. $M_2 \models_w (\alpha \vee \beta)$ *si y sólo si* $M_2 \models_w \alpha$ *o* $M_2 \models_w \beta$*;*

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

Semántica LMP 5. $M_2 \models_w M\alpha$ *si y sólo si, para algún* w' *t.q.* $(w, w') \in R$*,* $M_2 \models_{w'} \alpha$*;*

Una consecuencia de estas definiciones es que habrá tantos tipos diferentes de validación, como existen arreglos diferentes de mundos posibles, representados por la relación de accesibilidad. Estos diferentes tipos de validez, corresponden a diferentes **significados esperados** para los operadores modales. *Interpretaciones de los operadores*

Aunque el significado intuitivo de fbf válida en la LMP es el mismo que en su componente proposicional, su definición es ligeramente diferente al no ser funcionales los operadores modales. Una fbf es **válida** en el marco $\langle W, R \rangle$ si y sólo si, para cada modelo $\langle W, R, V \rangle$ basado en este marco, y para cada $w \in W$, $\alpha \in V(w)$. Existen algunas fbf que son válidas en cualquier arreglo de mundos posibles, y se conocen como **K-válidas**. Todas las fbf *Validez* *Fbf K-válidas*

4 Por Saul A. Kripke. La semántica de los mundos posibles fuer originalmente formulada por Hintikka [99] quien llamó a los mundos *alternativas epistémicas*, pero el desarrollo de los sistemas modales normales se debe a Kripke.

válidas de la lógica proposicional son K-válidas. En especial, una fbf modal que es K-válida es la siguiente: $L(p \to q) \to (Lp \to Lq)$.

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

1. Que todo teorema en el sistema es válido; y
2. Que toda fbf válida en el sistema es un teorema.

Consistencia y Completez

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

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

Equivalencia deductiva

Extensión propia

1.3.3 El sistema K

Los teoremas de este sistema son aquellas formulas que son fbf válidas en la lógica proposicional; y una sola fbf modal distintiva conocida como K [5]:

K 1 (LP-válida)**.** *Si α es una fbf válida de la Lógica Proposicional, entonces también lo es en K.*

K 2 (K)**.** $L(p \to q) \to (Lp \to Lq)$

Substitución uniforme

Y las siguientes reglas de inferencia: La regla de **substitución uniforme** (SU) establece que si una fbf α es un teorema; también lo será la fbf que resulta de substituir las variables proposicionales $p_1, \ldots, p_n$ en α, uniformemente por cualquier fbf $\beta_1, \ldots, \beta_n$:

Regla de Inferencia 4 (SU)**.**

$$\frac{\alpha}{\alpha[\beta_1/p_1, \ldots \beta_n/p_n]}$$

Ejemplo 1.17. *Como $p \vee \neg p$ es una fbf válida y por tanto un teorema. También lo es $(q \wedge r) \vee \neg(q \wedge r)$ que resulta de substituir la variable p por la fbf $(q \wedge r)$.*

Modus Ponens

La regla de **Modus Ponens**, o de eliminación de la implicación, expresa que si las fbf α y $\alpha \to \beta$ son teoremas, entonces también lo es β:

Regla de Inferencia 5 (MP)**.**

$$\frac{\alpha \quad \alpha \to \beta}{\beta}$$

5 Nuevamente, por Samuel Kripke.

Ejemplo 1.18. *Este patrón de razonamiento es bien conocido: Si comprar una pizza implica gastar dinero y compro una pizza, entonces gasto dinero.*

Las reglas anteriores no son propiamente modales. De hecho, la substitución uniforme es una regla plausible que se requiere para todo sistema lógico que incluya símbolos interpretados como variables proposicionales. El modus ponens simplemente refleja que el carácter funcional del operador implicación. La regla de **necesidad** (N) si que es una regla modal que preserva la validez con respecto a la definición de K-válido:

Necesidad

Regla de Inferencia 6 (N). $\vdash \alpha \rightarrow \vdash L\alpha$

Ejemplo 1.19. *Como* $\vdash (p \vee \neg p)$, *entonces* $\vdash L(p \vee \neg p)$.

Recuerden que las formulas K-válidas, son válidas para cualquier relación de accesibilidad. Intuitivamente, si α es proposicionalmente válido, α es válido en cualquier mundo posible, independientemente de la relación de accesibilidad entre los mundos. Para K, si el lado izquierdo de la implicación se cumple, tendría que haber un mundo donde Lq fuese falso, para que la implicación fuese falsa; pero dado que el lado izquierdo se cumple, ese mundo no es posible. Los teoremas derivados en K son K-válidos. Algunos teoremas en K son como sigue, su demostración detallada puede verse en Hughes y Cresswell [104]:

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

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

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

K 6. *Si* $\vdash \alpha \rightarrow \beta$ *entonces* $\vdash L\alpha \rightarrow L\beta$

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

K 8. *Si* $\vdash \alpha \equiv \beta$ *entonces* $\vdash L\alpha \equiv L\beta$

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

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

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

Substitución de equivalentes

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

Sistemas Modales Normales

1.3.4 El sistema T

Este sistema resulta al agregar el axioma $Lp \rightarrow p$ al sistema K. Como esta fbf no es K-válida, no es un teorema de K, lo que demuestra que el sistema T es diferente. T es aceptable para representar algunas nociones como –Lo que es necesario, es:

T 1 (T). $Lp \to p$

Otros teoremas de este sistema son:

T 2. $p \to Mp$

T 3. $M(p \to Lp)$

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

Marcos reflexivos

1.3.5 El sistema D

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

Interpretación Deóntica

D 1 (D). $Lp \to Mp$

Un teorema fácil de derivar en D es:

D 2. $M(p \to p)$

Observen que si α es un teorema en el sistema D, también lo es $M\alpha$. Cualquier sistema extensión de K que incluya algún teorema de la forma $M\alpha$, incluye al axioma D. T es una extensión propia de D y D lo es de K.

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

Marcos seriales

1.3.6 Los sistemas S4 y S5

Formulas como $Lp \to LLp$ deberían dejarnos perplejos. Contienen secuencias conocidas como **modalidades iteradas**, que nos llevan a preguntas como –¿Qué significa necesariamente necesario? $LLp \to Lp$ es una substitución de T, así que si agregamos $Lp \to LLp$ como un teorema, el nuevo sistema tendrá a $Lp \equiv LLp$ como teorema. Este tipo de teoremas se conoce como **leyes de reducción**. T no contiene ninguna regla de reducción. El sistema S4 se forma agregando el axioma 4 a K y T:

Modalidades iteradas

Leyes de reducción

S4 1 (4). $Lp \to LLp$

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

Forma estándar

Equivalencia

de remplazar una por otra en cualquier formula es siempre equivalente en el sistema, a la formula original.

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

Marcos transitivos

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

S5 1 (5). $Mp \rightarrow LMp$

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

Marcos bajo equivalencia

1.3.7 Teoría de correspondencia

Observen que los axiomas de los sistemas modales proposicionales, corresponden a ciertas propiedades de la relación de accesibilidad R. Esto se conoce como **Teoría de correspondencia** [104]. Para resumir, el Cuadro 1.2 muestra la teoría de correspondencia para las axiomas T, D, 4, y 5.

Teoría de correspondencia

A	Esquema	R	Cálculo de Predicados
T	$L\alpha \rightarrow \alpha$	reflexiva	$\forall w \in W,\ (w,w) \in R$
D	$L\alpha \rightarrow M\alpha$	serial	$\forall w \in W\ \exists w' \in W,\ (w,w') \in R$
4	$L\alpha \rightarrow LL\alpha$	transitiva	$\forall w,w',w'' \in W,\ (w,w') \in R \wedge (w',w'') \in R \rightarrow (w,w'') \in R$
5	$M\alpha \rightarrow LM\alpha$	euclidiana	$\forall w,w',w'' \in W,\ (w,w') \in R \wedge (w,w'') \in R \rightarrow (w',w'') \in R$

Cuadro 1.2: Teoría de correspondencia para los axiomas T,D,4, y 5.

Es posible construir $2^4 = 16$ diferentes sistemas modales proposicionales con estos cuatro axiomas[6], pero algunos de ellos son equivalentes. La figura 1.3 muestra la relación de inclusión entre los diferentes sistemas modales proposicionales normales, construidos con estos cuatro axiomas. Una flecha del sistema A al sistema B en el diagrama puede interpretarse como B es un superconjunto estricto de A, es decir, todo teorema de A es un teorema de B, pero no a la inversa. $A = B$ significa que A y B contienen exactamente los mismos teoremas. Recuerden que, como estos sistemas son ampliamente utilizados, se conocen como sigue: KT se conoce como T; $KT4$ se conoce como $S4$; $KD45$ se conoce como *weak-S5*; y $KT5$ se conoce como $S5$.

6 K es común a todos los sistemas.

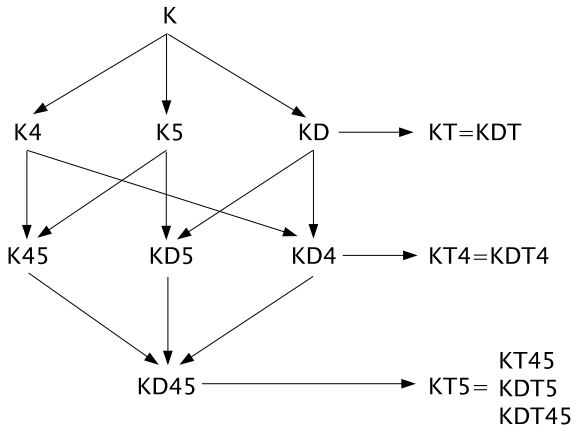

Figura 1.3: Sistemas modales normales, usando los axiomas T, D, 4 y 5.

1.4 LÓGICAS TEMPORALES

La interpretación temporal de los operadores modales ha sido central en las Ciencias de la Computación, particularmente en la especificación y verificación de programas vía la técnica conocida como **model checking** [37]. La idea es asumir una **Lógica Temporal** (LT), de forma que sea posible verificar si las propiedades deseables de nuestros programas, expresadas como fbf de esa lógica, se satisfacen siempre, o al menos eventualmente.

Model Checking
Lógica Temporal

A diferencia de la Lógica Proposicional, o la de Primer-Orden, donde el valor de verdad de las fbf es estático; en la LT, el valor de verdad de las fbf puede cambiar de un momento a otro. Ahora bien, diferentes concepciones sobre la **naturaleza del tiempo** dan lugar a diferentes lógicas. Desde esta tradición computacional de la LT, Emerson [55] define algunos aspectos a considerar en este sentido:

Naturaleza del tiempo

- **Proposicional** vs **Primer-Orden**. En realidad esta no es una consideración temporal. Se trata de decidir si las fbf sobre las cuales queremos razonar temporalmente son **proposicionales** o de **primer-orden**.
- **Global** vs **Modular**. Si los operadores temporales de la lógica se interpretan en un sólo universo de discurso, correspondiente a la ejecución de un sólo programa concurrente completo, se dice que la LT es **global**. Si los operadores se interpretan con respecto a diferentes fragmentos de un programa, se dice que la LT es **modular**.

  LT global
  LT modular
- **Lineal** vs **Arborescente**. Dada la naturaleza de los programas de cómputo, suele asumirse que para cada momento existe un único momento futuro posible, dando como resultado una estructura del tiempo **lineal**. Pero es posible considerar que para cada momento existe un conjunto de momentos futuros posibles, dando lugar a una estructura de tiempo **arborescente**.

  LT Lineal
  LT Arborescente
- **Momentos** vs **Intervalos.** Es posible adoptar operadores temporales para razonar acerca de las fbf en un **momento** dado en el tiempo; o bien sobre **intervalos** en él.

  LT Puntuales
  LT de Intervalos

- **Discreta** vs **Continua**. En la mayoría de las lógicas temporales, el tiempo se asume **discreto**, isomorfo a $\mathbb{N}$. Se puede asumir también que el tiempo es **continuo** o denso, isomorfo a $\mathbb{R}$. *LT Discretas* *LT continuas*

- **Pasado** vs **Futuro**. En sus orígenes en filosofía, las LT utilizaban modalidades temporales para describir la ocurrencia de eventos en el **pasado**. Sin embargo, en la mayoría de las lógicas temporales aplicadas al razonamiento sobre concurrencia en programas, sólo se utilizan modalidades sobre el **futuro**. *LT con Pasado* *LT con Futuro*

Observen que estas consideraciones se dan en el contexto de la aplicación de las lógicas temporales a la especificación y verificación de **programas reactivos** (en especial propiedades conocidas como *correctness* y *safeness*) [123, 37, 107]. Recuerden que los programas reactivos son aquellos que no tienen estados terminales definidos y su ejecución consiste en una interacción continua con el medio ambiente. Esta caracterización corresponde a las primeras definiciones que utilizamos sobre el término agente. *Programas reactivos*

En esta sección abordaremos a manera introductoria una LT Lineal Proposicional (LTLP) y, posteriormente, las dos lógicas temporales arborescentes utilizadas normalmente en los sistemas formales para describir agentes intencionales: *Computational Tree Logic* (CTL) y su extensión CTL* [56, 57, 55].

1.4.1 Lógica temporal proposicional lineal (LTPL)

En este sistema asumiremos una representación proposicional para las fbf no temporales y que la naturaleza del tiempo es lineal –Una sucesión de estados isomorfa a los números naturales $\mathbb{N}$.

Ejemplo 1.20. *Si queremos razonar acerca de si hay examen de lógica esta semana después de clase, necesitamos una representación como la que se muestra en la Figura 1.4. Hay un conjunto de momentos bien diferenciados, los días de la semana, isomorfa con $\mathbb{N}$. Un momento inicial, en este caso lunes, es asumido. Para cada momento hay una interpretación proposicional. Supongamos que p denota que hay clase de lógica y q que hay examen. Entonces es el caso que no hay clase lunes, lunes $\not\models p$; y hay examen el jueves, jueves $\models q$. Como veremos, también podremos expresar y probar que habrá examen al día siguiente de clase $\models \Diamond(p \rightarrow \bigcirc q)$.*

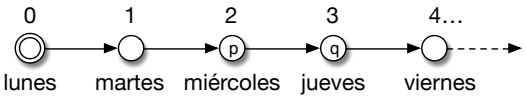

Figura 1.4: Una semana representada como una línea de tiempo discreta.

Sintaxis:

Los símbolos primarios de la LTPL son los siguientes:

VARIABLES PROPOSICIONALES: Var;

OPERADORES UNARIOS: $\neg$, $\bigcirc$;

OPERADORES BINARIOS: $\vee$, U ;

PARÉNTESIS: (,).

Sea $\mathcal{L}_{LTLP}$ el lenguaje de esta lógica, producido por las siguientes reglas sintácticas:

Sintaxis LTPL 1. *Si $\alpha \in Var$ entonces α es una fbf;*

Sintaxis LTPL 2. *Si α es una fbf, también lo son $\neg\alpha$ y $\bigcirc\alpha$;*

Sintaxis LTPL 3. *Si α y β son fbf, también lo son $\alpha \vee \beta$ y $\alpha \mathsf{U} \beta$*

Operadores temporales

El resto de los operadores lógicos son definidas como en la Lógica Proposicional. Con respecto a los **operadores temporales**, $\bigcirc$ es la abreviatura de en el siguiente momento (*next-time*), y U de hasta (*until*). La Figura 1.5 muestra el significado intuitivo de estos y otros operadores temporales como eventualmente y siempre:

Definición 1.12 (Eventualmente). $\Diamond\alpha =_{def} true\ \mathsf{U}\ \alpha$

Definición 1.13 (Siempre). $\Box\alpha =_{def} \neg(\Diamond\neg\alpha)$

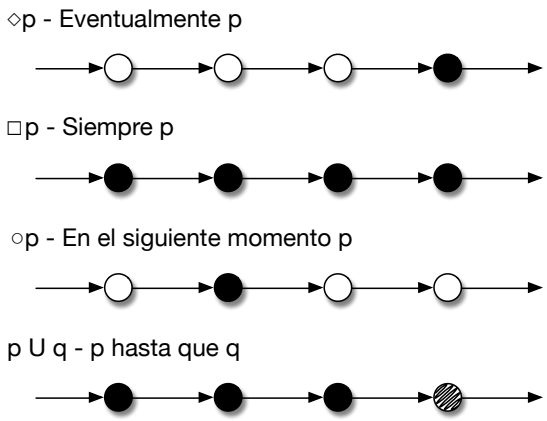

Figura 1.5: Significado intuitivo de los operadores temporales.

Ejemplo 1.21. *La expresión –Si envio un mensaje será recibido, se escribe:*

$$enviarMsj \rightarrow \Diamond recibirMsj$$

La expresión –Si alguien nace, entonces vive hasta su muerte, se escribe:

$$nacer \rightarrow vivir\ \mathsf{U}\ morir$$

Que eventualmente habrá examen al día siguiente de clase, se escribe:

$$\Diamond(clase \rightarrow \bigcirc examen)$$

Semántica

En esta lógica temporal, la estructura subyacente del tiempo el un conjunto totalmente ordenado $(S, <)$ que se asume isomorfo al conjunto de los números naturales con su orden usual $(\aleph, <)$. Bajo este supuesto, el tiempo es discreto y tiene un momento inicial sin predecesores, así como un futuro infinito. Estas propiedades parecen adecuadas para razonar acerca de las corridas de un programa representados como líneas de tiempo.

Definición 1.14 (Corrida). *Sea $M_3 = \langle S, x, L\rangle$, donde S es un conjunto de estados o **momentos**; $x : \aleph \to S$ es una secuencia posiblemente infinita de estados; y $L : S \to 2^{Var}$ es una interpretación para cada momento en S, donde Var es un conjunto de variables proposicionales. Entonces expresiones como $x = (s_0, s_1, s_2, \dots) = (x(0), x(1), x(2), \dots)$ se usan para denotar la línea de tiempo x. De manera alternativa, x se conoce como **trayecto** ó **corrida**.*

Momentos

Corridas

La expresión $M_3 \models_x \alpha$ significa "en la estructura M_3 la fórmula α es verdadera según la corrida x". Cuando se asume que M_3 es conocida, es posible escribir la expresión anterior como $\models_x \alpha$.

Las reglas semánticas de la LT lineal proposicional son:

Semántica LTPL 1. *$M_3 \models_x \alpha$, si y sólo si $\alpha \in L(x_0)$ para todo $\alpha \in Var$;*

Semántica LTPL 2. *$M_3 \models_x \neg\alpha$, si y sólo si $M_3 \not\models_x \alpha$;*

Semántica LTPL 3. *$M_3 \models_x \alpha \vee \beta$, si y sólo si $M_3 \models_x \alpha$ o $M_3 \models_x \beta$;*

Semántica LTPL 4. *$M_3 \models_x \alpha \,\mathsf{U}\, \beta$, si y sólo si $\exists j\ M_3 \models_{x_j} \beta$ y $\forall k < j\ M_3 \models_{x_k} \alpha$;*

Semántica LTPL 5. *$M_3 \models_x \bigcirc\alpha$, si y sólo si $M_3 \models_{x_1} \alpha$.*

Una fbf α se dice satisfacible si y sólo si existe una estructura lineal M, tal que $M \models_x \alpha$. Tal estructura define un modelo de α. Una fbf α se dice válida, si y sólo si para toda estructura lineal M, tenemos que $M \models_x \alpha$. Observen que α es válida, si y sólo si $\neg\alpha$ es no satisfacible.

Ejemplo 1.22. *$\alpha \to \Diamond\beta$ es satisfacible, pero no es válida. $\Box(\alpha \to \Diamond\beta)$ es satisfacible, pero no válida. $\Box(\alpha \to \Diamond\beta) \to (\alpha \to \Diamond\beta)$ es válida, pero su inverso es sólo satisfacible. $\alpha \wedge \Box(\alpha \to \bigcirc\alpha) \to \Box\alpha$ es válida y corresponde a la formulación de la inducción matemática.*

1.4.2 Lógica temporal proposicional arborescente (LTPA)

Como se mencionó, CTL y $CTL*$ son dos lógicas temporales arborescentes proposicionales. CTL permite operadores temporales básicos con la forma de cuantificadores sobre trayectos, ya sea A (inevitablemente o "en todo futuro") o E (opcionalmente o "en algún futuro"). En CTL estos cuantificadores anteceden a una sola combinación de los operadores temporales usuales: $\bigcirc$ (next), U (until), $\Diamond$ (sometime), y $\Box$ (always). En el caso de $CTL*$, los cuantificadores de trayectorias anteceden cualquier fbf de la lógica temporal lineal, incluyendo combinaciones booleanas usando los conectivos: $\wedge$, $\neg$, $\vee$, $\to$, $\equiv$.

Sintáxis LTAP

La sintaxis de estas lógicas arborescentes es casi igual a la de la lógica temporal lineal introducida en la sección anterior, la única diferencia es la introducción de cuantificadores de trayectorias:

VARIABLES PROPOSICIONALES: Var;

OPERADORES UNARIOS : $\neg$, $\bigcirc$;

OPERADORES BINARIOS: $\vee$, U ;

CUANTIFICADORES: A;

PARÉNTESIS: (,).

El cuantificador de trayectorias existencial (E) se define como:

Definición 1.15. $E\alpha =_{def} \neg A \neg \alpha$.

Otros operadores temporales como **eventualmente** (eventually) y **siempre** (always) se definen como sigue:

Definición 1.16 (eventually)**.** $\Diamond \alpha =_{def} true\ \mathsf{U}\ \alpha$

Definición 1.17 (always)**.** $\Box \alpha =_{def} \neg \Diamond \neg \alpha$

Ahora, es necesario establecer una categorización en la fbf de estas lógicas. Las **fórmulas de estado** se valúan a falso o verdadero con respecto a un estado del sistema, son aquellas que serán generadas por las reglas LTAP Sint 1 – 4. Las **fórmulas de trayectoria**, aquellas que toman su valor de verdad con respecto a una trayectoria, serán generadas por las reglas LTAP Sint 5 – 7:

Sintaxis LTPA 1. *Si $\alpha \in Var$, entonces α es una fbf de estado;*

Sintaxis LTPA 2. *Si α es una fbf de estado, también lo es $\neg\alpha$;*

Sintaxis LTPA 3. *Si α y β son fbf de estado, también lo es $(\alpha \vee \beta)$;*

Sintaxis LTPA 4. *Si α es una fbf de trayectoria, entonces $E\alpha$ y $A\alpha$ son fbf de estado;*

Sintaxis LTPA 5. *Si α es una fbf de estado, entonces también es una fbf de trayectoria;*

Sintaxis LTPA 6. *Si α es una fbf de trayectoria, también lo son $\neg\alpha$ y $\bigcirc\alpha$;*

Sintaxis LTPA 7. *Si α y β son fbf de trayectoria, también lo son $(\alpha \vee \beta)$ y $(\alpha\ \mathsf{U}\ \beta)$;*

Como es usual, los paréntesis externos pueden ser eliminados si esto no genera ambigüedad. El conjunto de fbf de estado generadas con estas reglas corresponde al lenguaje conocido como $CTL*$. La lógica temporal arborescente restringida, conocida como CTL se obtiene prohibiendo las combinaciones booleanas y los operadores temporales anidados al remplazar las reglas LTAP Sint 6 y LTAP Sint 7 por las siguientes reglas:

Sintaxis LTPA 8. *Si α es una fbf de estado, entonces $\bigcirc\alpha$ es una fbf de trayectoria;*

Sintaxis LTPA 9. *Si α y β son fbf de estado, entonces $\alpha\ \mathsf{U}\ \beta$ es una fbf de trayectoria.*

Semántica LTAP

En las lógicas temporales arborescentes proposicionales, la estructura subyacente del tiempo se asume de naturaleza arborescente, donde cada momento puede tener varios momentos sucesores posibles y un sólo predecesor. Tal estructura corresponde a un árbol infinito. En cada rama del árbol, la línea de tiempo correspondiente es isomorfa a $\aleph$. Normalmente, se permite que un nodo del árbol tenga infinitos sucesores y se requiere que al menos tenga uno. Por ello, los estados sin sucesores se convierten en estados reflexivos. Estos árboles, en el contexto de la discusión, son indistinguibles con respecto a los árboles con ramas finitas, y aún con ramas bien acotadas.

Una estructura temporal $M_4 = \langle S, R, L \rangle$ se define como sigue. S es un conjunto de estados. R es una relación binaria en $S \times S$ total, es decir, $\forall k \, \exists t \, (k, t) \in R \wedge (k \in R) \wedge (t \in R)$. La función $L : S \to 2^{Var}$ es un etiquetado (label), definido como en el cálculo proposicional. La figura 1.6 muestra una estructura de Kripke con interpretación temporal. W y R son representados directamente en la figura. L puede verse como la función que regresa las proposiciones verdaderas en cada mundo posible, por ejemplo, $L(w_1) = \{q, r, s\}$. El mundo w_1 representa el momento presente.

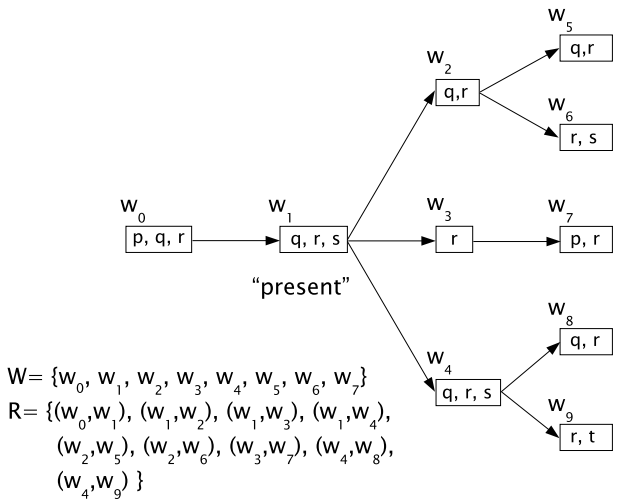

Figura 1.6: Un ejemplo de estructura temporal $\langle W, R, L \rangle$ para la LTAP, las letras en cada mundo posible w_i son las proposiciones que son verdaderas en ese mundo posible.

Las fbf de $CTL*$ se interpretan con respecto a la estructura M_4. Una corrida en esta estructura es una secuencia infinita $s_0, s_1, s_2, \ldots$ de mundos posibles o estados, tal que $(s_i, s_{i+1}) \in R$. Como se mencionó, $x = (s_0, s_1, s_2, \ldots)$ denota una corrida completa, mientras que x_i denota el sufijo de una trayectoria completa $(s_i, s_{i+1}, \ldots)$. $M_4 \models_x \alpha$ significa que la fbf de estado α es verdadera en el modelo M_4 con respecto al estado s_0 de la trayectoria x.

Las reglas semánticas de estas lógicas temporales arborescentes proposicionales son como siguen:

Semántica LTPA 1. *$M_4 \models_x \alpha$, si y sólo si $\alpha \in L(x_0)$ para todo $\alpha \in Var$;*

Semántica LTPA 2. *$M_4 \models_x \neg\alpha$, si y sólo si $M_4 \not\models \alpha$;*

Semántica LTPA 3. *$M_4 \models_x \alpha \vee \beta$, si y sólo si $M_4 \models_x \alpha$ o $M_4 \models_x \beta$;*

Semántica LTPA 4. $M_4 \models_x A\alpha$, *si y sólo si para toda trayectoria* $M \models \alpha$;

Semántica LTPA 5. $M_4 \models_x \alpha$, *si y sólo si existe una trayectoria completa* j *tal que* $M_4 \models_j \alpha$;

Semántica LTPA 6. $M_4 \models \alpha \cup \beta$, *si y sólo si* $\exists j\ M_4 \models_{x_j} \beta$ *and* $\forall k < j\ M_4 \models_{x_k} \alpha$;

Semántica LTPA 7. $M_4 \models_x \bigcirc\alpha$, *si y sólo si* $M_4 \models_{x_1} \alpha$.

La validez se define como de costumbre. Una fbf de estado α es **válida** si se cumple que para toda estructura M y todo estado s en toda trayectoria completa x, tenemos que $M \models \alpha$. Una fbf de estado α es **satisfacible** si existe al menos una estructura M que cumple estas condiciones.

Usando estas lógicas temporales es posible expresar propiedades acerca de la dinámica de un sistema, por ejemplo **viabilidad** (liveness) y **seguridad** (safely) [55]. La más sencilla de las propiedades de seguridad toma la forma de una invariante global, por ejemplo, una propiedad expresada por la fbf proposicional ϕ que es siempre verdadera: $\Box\phi$. Una invariante local puede expresar que siempre y cuando una condición expresada por ϕ sea cierta, la condición expresada por ψ también se cumple. Las invariantes locales toman la siguiente forma: $\Box(\phi \rightarrow \psi)$. La correctez parcial al final de la ejecución puede expresarse en términos de una precondición ϕ que es verdadera inicialmente, una postcondición ψ que debe ser verdadera al finalizar la ejecución, y una terminación χ indicando que el final dela ejecución ha sido alcanzado. La forma general de la correctez parcial es: $\phi \rightarrow \Box(\chi \rightarrow \psi)$. Las propiedades de viabilidad toman la forma $F\phi$ expresando que algo, ϕ en este caso, sucede eventualmente. La terminación es un ejemplo de propiedad de viabilidad, normalmente expresa que en toda trayectoria que inicialmente satisface una condición ϕ, eventualmente χ sucede, donde χ expresa las condiciones de terminación. La forma general de la terminación es $\psi \rightarrow \Diamond\chi$. Finalmente, la implicación temporal expresa que todo ϕ es seguido de ψ, su forma general es: $\Box(\phi \rightarrow \Diamond\psi)$.

1.4.3 Model checking en CTL

El problema de **model checking** [37] es fácil de describir. Dada una estructura de Kripke $M = (S, R, L)$ que representa un sistema concurrente de estados finitos; y una fórmula ϕ de alguna lógica temporal, que expresa alguna especificación deseada; encontrar el conjunto de todos los estados en S que satisfacen ϕ:

$$\{s \in S | M, s \models \phi\}$$

Normalmente, algunos estados del sistema concurrente se asumen como iniciales. El sistema satisface la especificación, si todos sus estados iniciales están en el conjunto de estados anterior.

Los primeros algoritmos para resolver este problema hacían uso de una representación explícita de la estructura de Kripke, como un grafo dirigido donde los nodos representan estados y los arcos están dados por la relación de accesibilidad R. Las estiquetas de los nodos están dadas por $L : S \rightarrow 2^{Var}$.

Para CTL [7] el algoritmo opera etiquetando cada estado s con el conjunto $label(s)$ de sub-fórmulas de ϕ que son verdaderas en s. Inicialmente, $label(s) = L(s)$. A partir de ahí, el algoritmo opera por varias etapas. En la etapa i-ésima, las fórmulas con $i-1$ operadores CTL anidados son procesadas. Cuando una sub-fórmula es procesada, es agregada a la etiqueta de aquellos estados donde es verdadera. Una vez que el algoritmo termina, tenemos que $M, s \models \phi$ si y sólo si $\phi \in label(s)$.

Recuerden que toda fórmula de CTL puede ser expresada en términos de $\neg$, $\vee$, $E\bigcirc$, $E\ \mathsf{U}$ y $E\Box$, por lo que para procesar las etapas intermedias del algoritmo, sólo es necesario procesar seis casos, dependiendo de si ϕ es atómica o tiene una de las siguientes formas: $\neg\phi$, $\phi \vee \psi$, $E \bigcirc \phi$, $E\ \phi\ \mathsf{U}\ \psi$, o $E \Box \phi$.

Para las fórmulas de la forma $\neg\phi$ etiquetamos aquellos estados que no están etiquetados con ϕ. Para $\phi \vee \psi$ etiquetamos aquellos estados etiquetados por ϕ o ψ. Para $E \bigcirc \phi$ etiquetamos aquellos estados que tienen un sucesor etiquetado por ϕ.

Para procesar las fórmulas $E\ \phi\ \mathsf{U}\ \psi$ primero encontramos todos los estados etiquetados con ψ. Después se trabaja hacía atrás usando el converso de la relación de accesibilidad R y encontramos todos los estados que pueden alcanzarse por un camino donde cada estado está etiquetado por ϕ. Todos esos estados deben etiquetarse con $E\ \phi\ \mathsf{U}\ \psi$. El algoritmo 1.1 muestra este procedimiento.

Algoritmo 1.1 Model Checking para fórmulas $E\ \mathsf{U}$

```
function checkEU(f1, f2)
    T ← {s | f2 ∈ label(s)};
    for all s ∈ T do
        label(s) ← label(s) ∪ {E f1 U f2};
    end for
    while T ≠ ∅ do
        s ← choose(T);
        T ← T \ {s};
        for all t s.t. R(t, s) do
            if E f1 U f2 ∉ label(t) ∧ f1 ∈ label(t) then
                label(t) ← label(t) ∪ {E f1 U f2};
                T ← T ∪ {t};
            end if
        end for
    end while
end function
```

El caso de las fórmulas $E\ \Box\phi$ es más complicado. Se basa en la descomposición del grafo en sus componentes no triviales fuertemente conectados. Un componente fuertemente conectado (SCC) C es un subgrafo maximal en el que todo nodo en C es alcanzable desde cualquier otro nodo en C a través de un camino dirigido contenido en C. C es no trivial si y sólo si contiene más de un nodo o contiene un nodo reflexivo.

Sea M' obtenida de M al borrar de S todos aquellos estados donde ϕ no ocurre, y restringir consecuentemente R y L. Por lo tanto $M' = (S', R', L')$

7 el proceso para $CTL*$ no es muy diferente, ver el libro de Clarke, Grumberg y Peled [37], capítulo 4.

donde $S' = \{s \in S | M, s \models \phi\}$, $R' = R_{|S' \times S'}$, y $L' = L_{S'}$. Observen que R' no será total en este caso. Los estados sin salida de transición pueden ser eliminados, pero este paso no es esencial para el funcionamiento del algoritmo.

El algoritmo depende de la siguiente observación: $M, s \models E \Box \phi$ si y sólo si dos condiciones se observan: i) $s \in S'$; y ii) Existe un camino en M' que lleva de s a un nodo t en un SCC del grafo (S', R'). El algoritmo 1.2 se sigue directamente de este lema.

Algoritmo 1.2 Model Checking para fórmulas EG

```
function checkEG(f_1)
    S' ← {s | f_1 ∈ label(s)};
    SCC ← {C | C es un SCC de S'};
    T ← ⋃_{C∈SCC} {s | s in C};
    for all s in T do
        label(s) ← label(s) ∪ {EG f_1};
    end for
    while T ≠ ∅ do
        s ← choose(T);
        T ← T \ {s};
        for all t s.t. t ∈ S' ∧ R(t, s) do
            if EG f_1 ∉ label(t) ∪ {EG f_1}; then
                label(t) ← label(t) ∪ {EG f_1};
                T ← T ∪ {t};
            end if
        end for
    end while
end function
```

La partición del grafo (S', R') en componentes fuertemente conectados se basa en al algoritmo de Tarjan [3]. Este algoritmo tiene una complejidad temporal $O(|S'| + |R'|)$. La computación completa del algoritmo se lleva a cabo en tiempo $O(|S| + |R|)$.

Para procesar una fórmula CTL ϕ, se aplica sucesivamente a las subfórmulas de ϕ, desde la fórmula más corta y más anidada, hacía el exterior. Como esto requiere $|\phi|$ pasadas, la complejidad total del algoritmo es $O(|\phi| \times (|S| + |R|))$.

El texto de Emerson [55] es también una referencia interesante sobre model checking y lógicas temporales.

1.5 LECTURAS Y EJERCICIOS SUGERIDOS

Huth y Ryan [107] ofrecen una excelente revisión de las lógicas desde el punto de vista de su uso en las ciencias de la computación. Fisher [64] ofrece una excelente introducción a los métodos formales basados en Lógicas Temporales.