## 1.4. Lógica de predicados

La lógica proposicional puede representar cualquier función booleana de un conjunto fijo de entradas booleanas, pero sus proposiciones atómicas no tienen estructura interna. Esto se vuelve incómodo cuando una afirmación depende de un dato o de un estado que cambia durante la ejecución de un programa. La misma limitación aparece cuando la validez de un argumento depende de la estructura interna de sus oraciones.

**Ejemplo.** Para expresar que una entrada $x$ es mayor que 3, no queremos inventar una proposición distinta para cada posible valor de $x$; queremos una sola expresión que dependa de $x$.

Consideremos los argumentos

| | |
|---|---|
| Todas las personas son mortales. | Si un usuario sigue a otro, puede ver sus publicaciones. |
| Sócrates es persona. | Alicia sigue a Bob. |
| Sócrates es mortal. | Alicia puede ver las publicaciones de Bob. |

Si cada oración se reemplaza por una proposición atómica, se pierde que "Sócrates" aparece en ambas premisas del primer argumento y que "Alicia" y "Bob" ocupan roles determinados en el segundo.

Nos falta un lenguaje que pueda hablar de objetos, de sus propiedades y relaciones, y que además permita cuantificar sobre ellos.

> **Definición.** La *lógica de predicados*, también llamada *lógica de primer orden*, extiende la lógica proposicional con objetos, funciones, predicados y cuantificadores.

**Ejemplo.** En el primer argumento anterior aparecen las propiedades "ser persona" y "ser mortal"; en el segundo aparece la relación "seguir a" entre dos usuarios. La lógica de primer orden permite representar explícitamente estas propiedades, relaciones y objetos compartidos.

### 1.4.1. Predicados y cuantificadores

Para que una misma afirmación pueda recibir distintos datos, necesitamos separar la propiedad que queremos comprobar de los objetos sobre los que la comprobamos.

> **Definición.** Fijado un *dominio de discurso* $D$, un *predicado* de aridad $n$ puede pensarse como una función $P\colon D^n \to \{0,1\}$. Un predicado de $n$ argumentos tiene *aridad* $n$.

**Ejemplo.** En el dominio $D = \mathbb{N}$, el predicado

$$P(x):\ \text{“}x > 3''$$

todavía no tiene un valor de verdad mientras $x$ no esté fijado. Al instanciarlo, $P(1)$ es falso y $P(5)$ es verdadero.

En un dominio de personas, los predicados unarios

$$\mathsf{Persona}(x), \qquad \mathsf{Mortal}(x)$$

representan propiedades, mientras que los predicados binarios

$$\mathsf{Sigue}(x,y), \qquad \mathsf{PuedeVer}(x,y)$$

representan relaciones entre dos objetos.

Instanciar $P$ en un objeto particular no basta para hablar de todo el dominio. Para expresar que una propiedad se cumple siempre o al menos una vez introducimos los cuantificadores.

**Definición.** El *cuantificador universal* $\forall$ forma la afirmación

$$\forall x\, \phi(x): \quad \text{“para todo objeto } x \text{ del dominio, se cumple } \phi(x)\text{”}.$$

**Ejemplo.** Si $D = \{d_1, \ldots, d_k\}$ es finito, entonces

$$\forall x\, \phi(x) \quad \text{equivale a} \quad \phi(d_1) \wedge \cdots \wedge \phi(d_k).$$

En el dominio infinito $\mathbb{Z}$, la fórmula

$$\forall x\, (x^2 \geq 0)$$

afirma que el cuadrado de todo entero es no negativo.

El cuantificador universal sirve para afirmar que no hay excepciones. Para afirmar que una propiedad se cumple al menos una vez necesitamos una segunda forma de recorrer el dominio.

**Definición.** El *cuantificador existencial* $\exists$ forma la afirmación

$$\exists x\, \phi(x): \quad \text{“existe al menos un objeto } x \text{ del dominio para el cual se cumple } \phi(x)\text{”}.$$

**Ejemplo.** Si $D = \{d_1, \ldots, d_k\}$ es finito, entonces

$$\exists x\, \phi(x) \quad \text{equivale a} \quad \phi(d_1) \vee \cdots \vee \phi(d_k).$$

En $\mathbb{Z}$, la fórmula $\exists x\, (x^2 = 4)$ es verdadera: 2 y $-2$ son posibles testigos. El cuantificador asegura que hay al menos uno, pero no que sea único.

### 1.4.2. Fórmulas y su sintaxis

Hasta ahora hemos usado el nuevo lenguaje de manera intuitiva. Para decidir qué expresiones están bien formadas, debemos distinguir primero los nombres de objetos de las afirmaciones que hacemos acerca de ellos.

**Definición.** Un *lenguaje de primer orden* $\mathcal{L}$ contiene:

- variables $x, y, z, \ldots$;
- símbolos de constante, que sirven para nombrar objetos;
- símbolos de función, cada uno con una aridad fija;
- símbolos de predicado, cada uno con una aridad fija;
- los conectivos lógicos, los cuantificadores $\forall$ y $\exists$, y los símbolos de puntuación.

**Ejemplo.** Un lenguaje para hablar de los números naturales puede contener una constante 0, una función unaria $s$ y un predicado binario $L$. Podemos pensar que $s$ representa la función sucesor y que $L(x, y)$ significa que $x < y$, aunque estos significados no forman parte de la sintaxis.

Los símbolos de un lenguaje son solamente las piezas disponibles. Antes de formar afirmaciones debemos precisar cuáles combinaciones de variables, constantes y funciones nombran objetos.

**Definición.** Los *términos* de un lenguaje $\mathcal{L}$ se definen recursivamente:

- toda variable es un término;
- todo símbolo de constante de $\mathcal{L}$ es un término;
- si $f$ es un símbolo de función de aridad $n$ y $t_1, \ldots, t_n$ son términos, entonces $f(t_1, \ldots, t_n)$ es un término.

**Ejemplo.** En el lenguaje anterior,

$$x, \qquad 0, \qquad s(0), \qquad s(s(x))$$

son términos. En cambio, $L(x, 0)$ no es un término: no nombra un objeto, sino que afirma algo acerca de dos objetos. Tampoco son términos $s$ ni $s(L(x, 0))$, pues $s$ requiere un argumento y ese argumento debe ser un término.

Los términos nombran objetos, pero todavía no expresan afirmaciones verdaderas o falsas. Necesitamos ahora reglas que indiquen cómo formar proposiciones a partir de ellos.

**Definición.** Las *fórmulas bien formadas* (FBF) de $\mathcal{L}$ se definen recursivamente:

- si $P$ es un símbolo de predicado de aridad $n$ y $t_1, \ldots, t_n$ son términos, entonces $P(t_1, \ldots, t_n)$ es una FBF;
- si $t$ y $u$ son términos, entonces $t = u$ es una FBF;
- si $\phi$ y $\psi$ son FBF, entonces $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$ y $(\phi \to \psi)$ son FBF;
- si $\phi$ es una FBF y $x$ es una variable, entonces $\forall x\, \phi$ y $\exists x\, \phi$ son FBF.

**Ejemplo.** Supongamos que $P$ es un predicado unario, $Q$ uno binario y $f$ una función unaria. Entonces

$$\forall x \left(P(f(x)) \to \exists y \left(Q(x, y) \wedge \neg P(f(z))\right)\right)$$

es una FBF. En cambio,

$$Q(\exists y\, P(y), x)$$

no lo es: $\exists y\, P(y)$ es una fórmula y aparece en una posición reservada para un término.

La gramática permite reconocer una FBF, pero no basta para saber qué ocurrencias de una variable controla cada cuantificador. Para ello debemos delimitar su alcance.

**Definición.** En $\forall x\,\phi$ y $\exists x\,\phi$, la fórmula $\phi$ es el *alcance* del cuantificador. Una ocurrencia de $x$ es *ligada* si está dentro del alcance de un cuantificador que la liga; en caso contrario es *libre*. Una fórmula sin variables libres se llama *fórmula cerrada*. Una fórmula con al menos una variable libre se llama *fórmula abierta*.

**Ejemplo.** En la fórmula

$$P(x) \wedge \exists x\, Q(x, y)$$

la primera ocurrencia de $x$ es libre, la segunda está ligada y $y$ queda libre. En la fórmula cerrada

$$\forall y \left(P(y) \to \exists x\, Q(x, y)\right)$$

todas las ocurrencias de variables están ligadas.

**Ejemplo.** El nombre de una variable ligada es irrelevante:

$$\exists x\, P(x) \vee \forall x\, Q(x) \quad \text{y} \quad \exists u\, P(u) \vee \forall v\, Q(v)$$

expresan lo mismo. Usaremos los criterios de precedencia de la lógica proposicional, delimitaremos con paréntesis todo alcance que contenga conectivos y abreviaremos $\forall x\,\forall y\, P(x, y)$ por $\forall x, y\, P(x, y)$.

Formalizaremos la noción de reemplazo.

**Definición.** Sea $\phi$ una FBF, $x$ una variable libre de ella y $t$ un término no ligado en $\phi$. El *reemplazo* de $x$ por $t$ en $\phi$, es la FBF obtenida reemplazando todas las ocurrencias libres de $x$ por $t$.

**Ejemplo.** Si $\phi(x)$ es

$$P(x) \to \exists y\, Q(x, y),$$

entonces $\phi(f(a))$ es

$$P(f(a)) \to \exists y\, Q(f(a), y).$$

Notemos que no podemos reemplazar $x$ por $y$ directamente en $\forall y\, P(x, y)$, pues la nueva ocurrencia de $y$ quedaría capturada. En cambio, no reemplazamos $x$ por $y$ en $\exists y\, Q(x, y)$, pues $y$ está ligada. Podemos renombrar primero la variable ligada y obtener $\exists z\, Q(y, z)$.

### 1.4.3. Cómo se interpretan las fórmulas

Una fórmula bien formada todavía no posee un valor de verdad por sí sola. Para interpretarla debemos especificar sobre qué objetos estamos hablando y qué significado tienen los símbolos del lenguaje.

**Definición.** Una *estructura* para un lenguaje de primer orden consiste en escoger un dominio de discurso no vacío y decidir qué objeto nombra cada constante, qué función representa cada símbolo de función y qué propiedad o relación representa cada símbolo de predicado.

**Ejemplo.** Consideremos un lenguaje con una constante 0 y una función unaria $s$. Si el dominio es $\mathbb{N}$, 0 nombra al cero y $s(n) = n + 1$, la fórmula cerrada

$$\exists x\, (s(x) = 0)$$

es falsa. Con la misma interpretación sobre el dominio $\mathbb{Z}$, es verdadera: podemos tomar $x = -1$. Así, una misma fórmula puede cambiar de valor de verdad cuando cambia la estructura.

Si una fórmula tiene variables libres, además debemos asignarles objetos del dominio. En $\mathbb{N}$, interpretemos $R(x, y)$ como "$x^2 = y$". La fórmula abierta $\exists x\, R(x, y)$ es verdadera si asignamos 4 a $y$, pero falsa si le asignamos 3.

Una estructura permite evaluar una fórmula en un escenario particular. Para comparar fórmulas entre todos los escenarios posibles necesitamos distinguir si funcionan en alguno o en todos ellos.

> **Definición.** Una fórmula es *satisfacible* si es verdadera para alguna estructura y alguna elección de valores para sus variables libres.

**Ejemplo.** La fórmula cerrada $\exists x\, (P(x) \vee \neg P(x))$ es válida porque los dominios son no vacíos. En cambio, $\exists x\, P(x)$ es satisfacible, pero no válida: podemos interpretar $P$ como una propiedad que algún objeto satisface o como una propiedad que ningún objeto satisface.

La validez describe una sola fórmula. Para estudiar argumentos necesitamos expresar cuándo la verdad de varias premisas obliga a que una conclusión también sea verdadera.

> **Definición.** Una fórmula $\psi$ es una *consecuencia lógica* de un conjunto de fórmulas $\Gamma$, lo que escribimos $\Gamma \models \psi$, si toda estructura y toda elección de valores que hacen verdaderas las fórmulas de $\Gamma$ hacen también verdadera a $\psi$.

**Ejemplo.** Se cumple

$$\{\forall x\, (P(x) \to Q(x)),\ P(a)\} \models Q(a),$$

porque en cualquier estructura la primera premisa se aplica, en particular, al objeto nombrado por $a$.

La definición recursiva completa que sustenta estas nociones se presenta en el apéndice de semántica formal.

### 1.4.4. Equivalencias con cuantificadores

Una misma condición cuantificada puede escribirse de varias maneras. Para reemplazar una forma por otra sin alterar un argumento necesitamos precisar cuándo ambas expresiones tienen exactamente el mismo significado.

> **Definición.** Dos FBF $\phi$ y $\psi$ son *lógicamente equivalentes*, lo que escribimos $\phi \equiv \psi$, si tienen el mismo valor de verdad en toda estructura y para toda elección de valores de sus variables libres.

**Ejemplo.** Se cumple $\forall x\, P(x) \equiv \forall y\, P(y)$: cambiar el nombre de una variable ligada no cambia el significado de la fórmula. Para refutar una equivalencia, en cambio, basta encontrar una estructura en la que las dos fórmulas tengan valores de verdad distintos.

Las equivalencias proposicionales de De Morgan y distribución sugieren preguntar cuáles de ellas siguen siendo válidas cuando aparecen cuantificadores. También queremos saber cuándo podemos cambiar el orden de dos cuantificadores.

**Proposición** (Equivalencias de cuantificadores)**.** Para $\phi(x)$ y $\psi(x)$ FBF:

$$\begin{aligned}
\neg\forall x\,\phi(x) &\equiv \exists x\,\neg\phi(x),\\
\neg\exists x\,\phi(x) &\equiv \forall x\,\neg\phi(x),\\
\forall x\,(\phi(x)\wedge\psi(x)) &\equiv \forall x\,\phi(x)\wedge\forall x\,\psi(x),\\
\exists x\,(\phi(x)\vee\psi(x)) &\equiv \exists x\,\phi(x)\vee\exists x\,\psi(x),\\
\forall x\,\forall y\,\phi(x,y) &\equiv \forall y\,\forall x\,\phi(x,y),\\
\exists x\,\exists y\,\phi(x,y) &\equiv \exists y\,\exists x\,\phi(x,y).
\end{aligned}$$

**Ejemplo.** Las dos primeras equivalencias son las leyes de De Morgan para cuantificadores. La negación de "todos los estudiantes entregaron la tarea" es "existe un estudiante que no la entregó". Análogamente, negar que exista un estudiante que faltó equivale a afirmar que ninguno faltó.

**Ejemplo.** No todas las distribuciones parecidas a las de la proposición son equivalencias. En el dominio $\mathbb{Z}$, sean $P(x)$: "$x$ es par" y $Q(x)$: "$x$ es impar". Todo entero es par o impar, pero no todos son pares ni todos son impares; por ello

$$\forall x\,(P(x)\vee Q(x)) \not\equiv \forall x\,P(x)\vee\forall x\,Q(x).$$

Además, existe un entero par y existe uno impar, pero ninguno es ambas cosas; por ello

$$\exists x\,P(x)\wedge\exists x\,Q(x) \not\equiv \exists x\,(P(x)\wedge Q(x)).$$

**Ejemplo.** Los cuantificadores distintos tampoco se pueden intercambiar libremente. En $\mathbb{R}$,

$$\forall x\,\exists y\,(x+y=0)$$

es verdadera porque, después de conocer $x$, podemos escoger $y=-x$. En cambio, $\exists y\,\forall x\,(x+y=0)$ es falsa: exigiría un único valor de $y$ que funcionara para todos los reales. Por lo tanto, ambas fórmulas no son equivalentes.

# Ejercicios

1. El dominio contiene personas y computadores. Use los predicados

$$E(x):\ \text{“}x\text{ es estudiante de esta clase”},\qquad C(x):\ \text{“}x\text{ es un computador”},$$

y $H(x,y)$: "$x$ ha usado $y$".

   (a) Traduzca las siguientes afirmaciones a FBF de primer orden:
      (I) todo estudiante de esta clase ha usado algún computador;
      (II) existe un computador que todos los estudiantes de esta clase han usado;
      (III) ningún estudiante de esta clase ha usado todos los computadores;
      (IV) cada computador ha sido usado por a lo más un estudiante de esta clase.

   (b) Construya una estructura finita en la que la afirmación (i) sea verdadera y la afirmación (ii) sea falsa. Explique qué muestra su ejemplo sobre el orden de los cuantificadores.

(c) En

$$P(x) \wedge \forall x \, \big(Q(x,y) \to \exists z \, R(y,z)\big),$$

identifique las ocurrencias libres y ligadas de cada variable, y decida si la fórmula es abierta o cerrada.

(d) Escriba la negación de las afirmaciones (i) y (ii) como FBF en las que la negación aparezca solamente delante de fórmulas atómicas.

2. Un lector de strings recibe un mensaje $w = w_1 \cdots w_n$, escrito como una palabra no vacía sobre el alfabeto $\{a, b\}$. Podemos ver el mensaje como una estructura $\mathcal{M}_w$ cuyo dominio es el conjunto de posiciones $\{1, \ldots, n\}$ y cuyo lenguaje contiene los predicados $A$, $B$ y $<$, interpretados por

$$A(i) \iff w_i = a, \qquad B(i) \iff w_i = b,$$

y por el orden usual entre las posiciones. Así, las variables representan posiciones de la palabra, no sus letras.

(a) Construya fórmulas abiertas $\mathsf{Primera}(x)$, $\mathsf{Ultima}(x)$ y $\mathsf{Siguiente}(x, y)$ que expresen, respectivamente, que $x$ es la primera posición, que $x$ es la última posición y que $y$ es la posición inmediatamente posterior a $x$.

(b) Usando las fórmulas del inciso anterior como abreviaturas, construya una FBF cerrada para cada una de las siguientes propiedades de $w$:

(I) la primera y la última letra son iguales;

(II) cada aparición de $a$ está seguida inmediatamente por una $b$;

(III) todas las apariciones de $a$ están antes que todas las apariciones de $b$;

(IV) el bloque consecutivo $ab$ aparece exactamente una vez.

(c) Determine cuáles de las cuatro fórmulas construidas son verdaderas en cada una de las palabras

$$\texttt{abba}, \qquad \texttt{aabb}, \qquad \texttt{abab}.$$

(d) Describa exactamente cuáles palabras satisfacen simultáneamente las propiedades (ii) y (iii), y justifique su respuesta.

3. Consideremos estructuras para un lenguaje con un predicado binario $E$. Podemos ver cada estructura como una red dirigida: $E(x, y)$ indica que $x$ puede enviar directamente un mensaje a $y$. Ampliamos el lenguaje con constantes $s, t$ que distinguen dos vértices. Una *cadena dirigida de longitud $n$* de $s$ a $t$ es una sucesión $s = u_0, u_1, \ldots, u_n = t$ tal que $E(u_i, u_{i+1})$ para todo $0 \le i < n$.

La *alcanzabilidad* de $t$ desde $s$ afirma que existe una cadena dirigida de alguna longitud positiva, sin fijar esa longitud de antemano. Demostraremos que ninguna FBF de primer orden expresa la alcanzabilidad en todas las redes dirigidas.

(a) Para cada $n \ge 1$, construya una FBF cerrada $\varphi_n(s, t)$ que sea verdadera exactamente cuando existe una cadena dirigida de longitud $n$ de $s$ a $t$.

(b) Suponga, buscando una contradicción, que existe una fórmula $\psi(x, y)$, en el lenguaje que contiene solamente a $E$, que expresa la alcanzabilidad en todas las redes dirigidas. Considere el conjunto

$$\Gamma = \{\psi(s, t)\} \cup \{\neg\varphi_n(s, t) : n \geq 1\}.$$

Demuestre que $\Gamma$ no es satisfacible.

(c) Demuestre que todo subconjunto finito $\Delta \subseteq \Gamma$ es satisfacible.

(d) Use (sin demostrar) que $\Gamma$ es satisfacible si y solo si todos sus subconjuntos finitos son satisfacibles para concluir que la alcanzabilidad no puede expresarse mediante ninguna FBF de primer orden.