transparent

BLOQUE II:

Tema 1

SINTAXIS DE LA LÓGICA PROPOSICIONAL

Lógica

Grado en Ingeniería Informática

Alessandra Gallinari

URJC

Contenido

Definiciones generales

Como ya comentamos en el capítulo de introducción, la **sintaxis** es la definición axiomática de los elementos básicos del lenguaje y de las reglas que permiten obtener nuevas expresiones correctas a partir de aquellos, las fórmulas.
Recordamos que se puede fijar el origen de la lógica matemática (y de la lógica proposicional) al final del siglo XIX, coincidiendo con la aparición de las obras de G. Boole (1815-1864) y de G. Frege (1848-1925).
El objetivo de este capítulo es el estudio de la sintaxis de la **lógica proposicional** que nos permite analizar las fórmulas proposicionales construidas a partir de fórmulas atómicas (proposiciones declarativas simples) y conectivos lógicos.

- Un **alfabeto** A es un conjunto de símbolos.
- Una **palabra sobre el alfabeto** A es una secuencia finita de símbolos de A. Al conjunto de todas las posibles palabras se le suele denotar como A^*.
- Un **lenguaje sobre el alfabeto** A es un cualquier subconjunto de A^*.
- Las **reglas de formación** son las reglas que permiten obtener nuevas expresiones de un lenguaje a partir de expresiones básicas.

Definiciones generales

Así, por ejemplo, si A es el alfabeto del idioma español, "aytzco" es una palabra sobre A.
Un lenguaje es el conjunto de las palabras del diccionario de español y la gramática española define las reglas de formación de expresiones correctas.
A continuación vamos a definir el lenguaje de la lógica de proposiciones por medio de su alfabeto y sus reglas de formación.

Alfabeto de la lógica proposicional

Los elementos básicos del **alfabeto del la lógica proposicional** son:

- **Las proposiciones atómicas (enunciados simples o variables proposicionales):** son proposiciones (oraciones declarativas a las cuales se pueden asociar valores de verdad) que no pueden descomponerse en otras proposiciones más simples.
  Para representar las proposiciones atómicas se suelen usar los símbolos p, q, r, s, $t, \ldots$ El conjunto de estos símbolos se suele denominar **signatura**.

Ejemplos

Los siguientes son ejemplos de proposiciones simples:
$p =$ *la raíz cuadrada de 2 es irracional,*
$q =$ *hoy me siento feliz,*
$t =$ *los gatos son felinos.*

Conectivos

- **Los conectivos lógicos:**
  - **constantes (de aridad 0):** $\top$ (verdadero) y $\bot$ (falso)
  - **conectivos unarios (de aridad 1):** $\neg$ (negación)
  - **conectivos binarios (de aridad 2):**
    $\wedge$ (y: la conjunción),
    $\vee$ (ó: la disyunción),
    $\rightarrow$ (la implicación o condicional),
    $\leftrightarrow$ (la doble implicación o coimplicación).

    NOTACIÓN: El símbolo $\circ$ se usará para representar un conectivo binario cualquiera.

Conectivos

Ejemplo

Siendo q = hoy me siento feliz, aplicando la negación se obtiene $\neg(q)$ = hoy no me siento feliz.

Ejemplos

1) La frase "Hoy llueve, sin embargo no hace frío" se escribe como $(p \wedge \neg(q))$, donde p = hoy llueve y q = hace frío.
2) La frase "Los lápices de mi hermana son rojos o negros" se escribe como $(p \vee q)$, donde p = los lápices de mi hermana son rojos y q = los lápices de mi hermana son negros.
3) La frase "La luna es de papel si y sólo si Carlos lee muchos libros" se escribe como $(p \leftrightarrow q)$, donde p = la luna es de papel y q = Carlos lee muchos libros.

Tabla de los conectivos

Conectiva lingüística	Conectivo lógico	Símbolo	Fórmula
verdadero	*Constante (de aridad 0)*	$\top$	$\top$
falso	*Constante (de aridad 0)*	$\bot$	$\bot$
no p	*Negación (unario)*	$\neg$	$\neg(p)$
p **y** q	*Conjunción (binario)*	$\wedge$	$(p \wedge q)$
p **ó** q	*Disyunción (binario)*	$\vee$	$(p \vee q)$
si p **entonces** q	*Implicación (binario)*	$\rightarrow$	$(p \rightarrow q)$
p **si y sólo si** q	*Coimplicación (binario)*	$\leftrightarrow$	$(p \leftrightarrow q)$ o $(p \rightarrow q) \wedge (q \rightarrow p)$

Cuadro: Los conectivos de la lógica proposicional

Símbolos de puntuación

- **Los símbolos de puntuación (o símbolos auxiliares):** paréntesis abiertos y cerrados.

Ejemplo

La frase "Si n es un número primo y mayor que 2, entonces n es impar" se escribe como $((p \wedge q) \rightarrow r)$, *donde* $p =$ *n es primo,* $q =$ *n es mayor que 2 y* $r =$ *n es impar.*

Definiciones recursivas

Los elementos básicos de un lenguaje permiten definir cadenas finitas de símbolos arbitrarias (**palabras**). Así, por ejemplo, la palabra $()\neg p \wedge \vee)q \to)$ se puede formar a partir del alfabeto de la lógica de proposiciones.
De todas las posibles palabras, nos interesa definir aquellas que se obtienen a partir del alfabeto dado sólo por medio de la reglas de formación de nuestro lenguaje.

En matemáticas y en la ciencia de la computación a menudo se usan **definiciones recursivas** de conjuntos, funciones o algoritmos, como en los siguientes ejemplos:

Definiciones recursivas

Ejemplos

1) Para definir el conjunto $\mathbb{N} = \{1, 2, 3, 4, \dots\}$ de los números naturales se puede dar un símbolo inicial 1 y unas reglas de formación, que nos permiten hallar el resto de los elementos del conjunto.

Axiomas de Peano (siglo XIX):

- *En $\mathbb{N}$ hay un elemento distinguido que denominamos* $\mathbf{1}$.
- *Para cada $n \in \mathbb{N}$ se define de manera única el* **siguiente** *de n. El siguiente de n se denota $s(n) = n + 1$ y es un elemento de $\mathbb{N}$ para cada $n \in \mathbb{N}$.*
- *No existe ningún número natural n tal que $s(n) = 1$.*
- *Si $s(n) = s(m)$ entonces $n = m$.*
- **Principio de inducción**: *si un conjunto de números naturales contiene al 1 y a los sucesores de cada uno de sus elementos, entonces contiene a todos los números naturales.*

Definiciones recursivas

Ejemplos

2) Otro ejemplo muy conocido es la definición recursiva de la sucesión $\{f_n\}_{n\in\mathbb{N}}$ *de Fibonacci (hacia* $1175 - 1250$*), que fue definida para estudiar la reproducción de los conejos. Sus primeros dos términos son* $f_1 = 1$ *y* $f_2 = 1$*. Si* $n \geq 3$*, entonces el valor* f_n *se deduce de los valores* f_{n-1} *y* f_{n-2} *según la fórmula* $f_n = f_{n-1} + f_{n-2}$*. Se sigue que* $\{f_n\} = \{1, 1, 2, 3, 5, 8, 13, \cdots\}$.
3) El factorial de un número natural n,

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 2 \cdot 1,$$

se puede definir también de forma recursiva:

$$n! = \begin{cases} 1 & si \quad n = 1, \\ n(n-1) & si \quad n > 1. \end{cases}$$

Definiciones recursivas

Ejemplos

4) El algoritmo de búsqueda binaria es un ejemplo de algoritmo de tipo "divide y vencerás," que se define de forma recursiva. El problema que se quiere resolver es de determinar si un cierto número x *pertenece a una dada lista (finita) ordenada de números* l*. El algoritmo de búsqueda binaria divide primero la lista* l *en dos mitades y compara* x *con un elemento central. Si el elemento central es distinto de* x*, determina a cuál de las mitades puede pertenecer* x*. Seleccionada esta nueva lista, vuelve a comparar* x *con su elemento central y a seleccionar aquella mitad de la nueva lista que puede contener* x*. Estas particiones se repiten hasta llegar a una lista de un sólo elemento, que puede ser o no ser igual a* x*.*

Definición recursiva de las fórmulas proposicionales

Definición

Una **fórmula proposicional (fórmula bien construida, fbc)** *es una palabra sobre el alfabeto de la lógica proposicional que puede construirse en un número finito de pasos mediante las reglas de formación que vamos a definir a continuación:*

1. *(At): Los símbolos* $\top$ *y* $\bot$ *y toda proposición atómica son una fórmula.*
2. *($\neg$): Si* φ *es una fórmula entonces* $\neg\varphi$ *es una fórmula.*
3. *($\circ$): Si* φ *y* ψ *son dos fórmulas entonces* $(\varphi \circ \psi)$ *es una fórmula.*
4. *Si una palabra no se obtiene mediante las tres reglas anteriores entonces no es una fórmula.*

NOTACIÓN: Para representar las fórmulas se suelen usar las letras del alfabeto griego $\varphi, \psi, \chi, \ldots$ (ver la tabla del alfabeto griego).

Subfórmulas

Definición

Dadas dos fórmulas φ *y* ψ*, se dice que* ψ *es una* **subfórmula** *de* φ *si* ψ *consiste de símbolos consecutivos de* φ*. En particular, cada fórmula es una subfórmula de sí misma.*

Ejemplo

La palabra $((p \wedge q) \to r)$*, del ejemplo (2) es una fórmula bien construida ya que:*
1. p, q, r*, son fórmulas atómicas (aplicando (At)),*
2. $(p \wedge q)$ *es una fórmula proposicional (aplicando ($\circ$)),*
3. $((p \wedge q) \to r)$ *es una fórmula proposicional (aplicando ($\circ$)).*

Además, $(p \wedge q)$ *y* r *son subfórmulas de* $((p \wedge q) \to r)$*.*

Inducción estructural: demostración de propiedades

Para poder **estudiar propiedades** de las fórmulas proposicionales una de las técnicas más adecuadas es el **principio de inducción estructural para fórmulas proposicionales**, que es una versión generalizada (y más compleja) del razonamiento de inducción definido a partir de los axiomas de Peano de los números naturales.
Antes de enunciar el principio de inducción estructural para fórmulas proposicionales, recordamos la definición de razonamiento por inducción sobre los números naturales:

Inducción sobre $\mathbb{N}$

Sea $P(n)$ una propiedad para números naturales. Supongamos que se pueda probar que:

1. **Base de inducción:** se verifica $P(1)$,
2. **Paso de inducción:** si se verifica $P(n)$ (**hipótesis de inducción**), entonces se verifica $P(n+1)$.

Bajo las hipótesis anteriores, se sigue que se verifica $P(n)$ para **todo número natural** n.

La demostración de la validez del razonamiento de inducción se obtiene aplicando el principio de inducción al conjunto
$A = \{n \in \mathbb{N} : P(n) \text{ se verifica}\}$.

Inducción sobre $\mathbb{N}$

Ejemplo

Vamos a demostrar por inducción que para todo $n \in \mathbb{N}$,

$$2^n \leq (n+1)!$$

Base de inducción: $P(1)$ *es verdadera, siendo* $(2 \leq 2)$.
Paso de inducción: si $2^n \leq (n+1)!$ *(si se verifica* $P(n)$*), entonces*

$$2^{n+1} = 2\ 2^n \leq 2\ (n+1)! \leq (n+2)\ (n+1)! = (n+2)!.$$

Por tanto, $2^n \leq (n+1)!$ *se verifica para todo número natural* n.

Inducción estructural

El principio de inducción estructural generaliza el método de demostración por inducción. Ese método de demostración se emplea para verificar propiedades de conjuntos generados inductivamente (recursivamente), cuyos elementos se obtienen como resultado de aplicar un número finito de reglas de formación.
Sus aplicaciones son numerosas y en informática se utiliza, por ejemplo, para la verificación de programas.

Inducción estructural para fórmulas proposicionales

Sea $\mathbb{P}$ una cierta propiedad tal que:

1. **Base de inducción (At):** Los símbolos $\top$ y $\bot$ y toda proposición atómica cumplen la propiedad $\mathbb{P}$.
2. **Pasos de inducción:**
   - $(\neg)$: Si φ es una fórmula que cumple la propiedad $\mathbb{P}$ (hipótesis de inducción), entonces $\neg(\varphi)$ cumple la propiedad $\mathbb{P}$.
   - $(\circ)$: Si φ y ψ son dos fórmulas que cumplen la propiedad $\mathbb{P}$ (hipótesis de inducción), entonces $(\varphi \circ \psi)$ cumple la propiedad $\mathbb{P}$.

Entonces, el **principio de inducción estructural para fórmulas proposicionales** afirma que, si se verifican las condiciones anteriores, se puede concluir que **toda fórmula proposicional** cumple la propiedad $\mathbb{P}$.

Inducción estructural para fórmulas proposicionales

Ejemplo

Usando el principio de inducción estructural podemos probar que toda fórmula φ contiene el mismo número de paréntesis abiertos y cerrados:
Base de inducción (At): *Los símbolos $\top$ y $\bot$ y toda proposición atómica contienen 0 paréntesis abiertos y 0 paréntesis cerrados.*
Pasos de inducción:
$(\neg)$: Si φ es una fórmula que contiene n paréntesis abiertos y n paréntesis cerrados, entonces $\neg(\varphi)$ contiene $n+1$ paréntesis abiertos y $n+1$ paréntesis cerrados.
$(\circ)$: Si φ y ψ son dos fórmulas que contienen n_φ y n_ψ paréntesis abiertos y n_φ y n_ψ paréntesis cerrados respectivamente, entonces $(\varphi \circ \psi)$ contiene $n_\varphi + n_\psi + 1$ paréntesis abiertos y $n_\varphi + n_\psi + 1$ paréntesis cerrados.

Se sigue que, por el principio de inducción estructural, toda fórmula φ contiene el mismo número de paréntesis abiertos y cerrados.

Representación de las fórmulas. Forma usual

En este apartado vamos a ilustrar las maneras más usadas para representar las fórmulas de la lógica proposicional.

Forma usual

Siguiendo las reglas de formación de las fórmulas proposicionales obtenemos ejemplos de expresiones bien construidas como las siguientes:

1. $((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t))$,
2. $(p \wedge (\neg(\neg((q \rightarrow r) \rightarrow (p \vee (\neg(r)))))))$,
3. $(p \rightarrow (q \rightarrow r))$.

Se puede notar que se usan paréntesis para evitar ambigüedad en la formalización. Sin embargo este uso se puede relajar obteniendo expresiones que no son fórmulas bien construidas, pero vienen empleadas como tales por razones de convenios.

Forma abreviada

Forma abreviada

Para reducir una fórmula bien construida a su **forma abreviada** podemos seguir los siguientes pasos:

a) Se puede **omitir el par de paréntesis externo**.
Así, por ejemplo, las anteriores fórmulas 1., 2. y 3. se escribirían como:
1.1. $(p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$,
2.1. $p \wedge (\neg(\neg((q \rightarrow r) \rightarrow (p \vee (\neg(r))))))$,
3.1. $p \rightarrow (q \rightarrow r)$.

b) Se introducen las siguientes **reglas de precedencia** entre conectivos que definen las prioridades que tenemos que respetar a la hora de aplicarlos:

Nivel 1: $\neg$,
Nivel 2: $\vee$ y $\wedge$,
Nivel 3: $\rightarrow$ y $\leftrightarrow$.

Forma abreviada

Así, por ejemplo, las anteriores fórmulas 1.1, 2.1. y 3.1 se escribirían como:
1.2. $(p \wedge (p \to r)) \vee (q \leftrightarrow t)$,
2.2. $p \wedge \neg\neg((q \to r) \to p \vee \neg r)$,
3.2. $p \to (q \to r)$.

c) Se admite el **convenio de asociatividad**: los conectivos $\vee$, $\wedge$, $\to$ y $\leftrightarrow$ se asocian por la derecha:

$$p \wedge q \wedge r \quad \text{es} \quad p \wedge (q \wedge r)$$

$$p \vee q \vee r \quad \text{es} \quad p \vee (q \vee r)$$

$$p \to q \to r \quad \text{es} \quad p \to (q \to r).$$

Así, por ejemplo, las anteriores fórmulas 1.2, 2.2. y 3.2 se escribirían como:
1.3. $(p \wedge (p \to r)) \vee (q \leftrightarrow t)$,
2.3. $p \wedge \neg\neg((q \to r) \to p \vee \neg r)$,
3.3. $p \to q \to r$.

Principio de unicidad de estructura

El siguiente **principio de unicidad de estructura para fórmulas proposicionales** afirma que cada fórmula admite un análisis sintáctico único, es decir, existe una única manera de derivar una fórmula usando las reglas de formación dadas.

Principio de unicidad de estructura para fórmulas proposicionales

Toda fórmula proposicional φ pertenece a una y sólo una de las siguientes categorías:

1. (At): φ es atómica,
2. $(\neg)$: φ es $\neg(\varphi_1)$ para cierta fórmula φ_1,
3. $(\circ)$: φ es $(\varphi_1 \circ \varphi_2)$ para cierto conectivo $\circ$ y ciertas fórmulas φ_1 y φ_2.

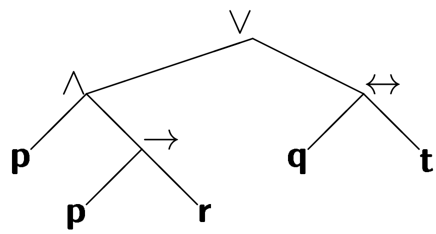
Además, en todos los casos la fórmulas φ_1 y φ_2 están unívocamente determinadas.

Una primera consecuencia del principio de unicidad de estructura es la posibilidad de representar fórmulas proposicionales por medio de árboles.

Árboles sintácticos

Ejemplo

Sea φ *la fórmula proposicional* $((p \wedge (p \to r)) \vee (q \leftrightarrow t))$.
Para representar la estructura sintáctica de φ, *podemos dibujar el árbol con raíz de la figura.*

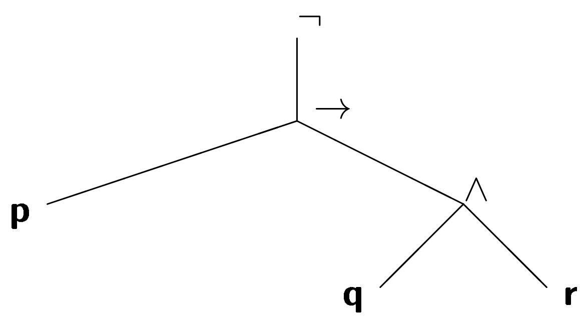


Las hojas del árbol obtenido representan las fórmulas atómicas a partir de las cuales se ha generado φ. *El árbol se lee de abajo hacía arriba y en cada nodo interior y en la raíz aparecen los conectivos empleados en la formación de nuestra fórmula.*

Árbol sintáctico

Ejemplo

La figura representa el árbol sintáctico de la fórmula $(\neg(p \to (q \wedge r)))$.



Principio de recursión estructural: definición de funciones

Una segunda consecuencia del principio de unicidad de estructura es el **principio de recursión estructural para fórmulas proposicionales** que nos permite **definir funciones**

$$f : \mathbf{L} \longrightarrow \mathbf{A}$$

cuyo dominio sea el conjunto **L** de todas las fórmulas proposicionales y cuyo codominio sea un conjunto dado **A**.

El principio de recursión estructural para fórmulas proposicionales consiste en la siguiente definición recursiva de la función $f : \mathbf{L} \longrightarrow \mathbf{A}$:

1. **Base (At):** Si φ es atómica, $f(\varphi)$ se define explícitamente, es un elemento de **A**,
2. **Pasos recursivos:**
    - $(\neg)$: $f(\neg(\varphi))$ es un valor que depende de $\neg$ y de $f(\varphi)$,
    - $(\circ)$: $f(\varphi_1 \circ \varphi_2)$ para cierto conectivo $\circ$ y ciertas fórmulas φ_1 y φ_2 es un valor que depende de $\circ$, $f(\varphi_1)$ y $f(\varphi_2)$.

Estas definiciones determinan la función f sobre todo **L**.

Principio de recursión estructural: definición de funciones

Ejemplo (HLR)

Se puede definir recursivamente la función

$$f : \mathbf{L} \longrightarrow \mathbb{N} \cup \{\mathbf{0}\}$$

que a cada fórmula $\varphi \in \mathbf{L}$ asocia el número (entero no negativo) de conectivos binarios en la estructura sintáctica de φ :

1. **Base (At):** *Si φ es atómica,* $f(\varphi) = 0$
2. **Pasos recursivos:**
   $(\neg)$: $f(\neg(\varphi)) = f(\varphi)$
   $(\circ)$: $f(\varphi_1 \circ \varphi_2) = f(\varphi_1) + f(\varphi_2) + 1$, *para cierto conectivo $\circ$ y ciertas fórmulas φ_1 y φ_2.*

Estas definiciones determinan la función f sobre todo **L**.

Tabla de formalización

La formalización es una herramienta básica y el alumno tendrá ocasión de practicarla a lo largo de todo el estudio de la lógica proposicional y de predicados.

Expresiones en el lenguaje natural	Formalización
no p **es falso que** p **no es cierto** p	$\neg p$
p **y** q p **pero** q p **sin embargo** q p **no obstante** q p **a pesar de** q	$p \wedge q$
o p **o** q **o ambas cosas** **al menos** p **o** q **como mínimo** p **o** q	$p \vee q$
si p **entonces** q p **sólo si** q q **si** p q **es necesario para** p p **es suficiente para** q **no** p **a menos que** q **no** p **o** q	$p \to q$
p **si y sólo si** q p **es necesario y suficiente para** q q **es necesario y suficiente para** p	$p \leftrightarrow q$ **o** $(p \to q) \wedge (q \to p)$

Formalización

Para formalizar un **razonamiento** con premisas $p_1, p_2, \ldots, p_n$ y conclusión q usaremos cualquiera de las dos formas:

$$\begin{array}{l} p_1 \\ p_2 \\ \vdots \\ p_n \\ \hline q \end{array} \qquad o \qquad p_1 \wedge p_2 \wedge \cdots \wedge p_n \to q.$$

Formalización

Observación

Una manera sencilla de verificar la validez de la formalización de una frase del lenguaje natural es de volver a traducir al lenguaje natural la formalización obtenida.
Así, por ejemplo, consideremos la frase "No voy a la playa a menos que haga mucho calor."
Siendo $p =$ voy a la playa y $q =$ hace mucho calor, la formalización $(p \to q)$ es correcta y la formalización $(q \to p)$ no es correcta.
En efecto, la primera se lee como "Voy a la playa sólo si hace mucho calor" y la segunda como "Si hace mucho calor, entonces voy a la playa."
En la segunda formalización se ha intercambiado la condición necesaria (la conclusión) con la suficiente (la premisa).

Formalización

Ejemplos

1) El enunciado "Si una función f es derivable en el intervalo $[a, b]$, entonces f es continua en $[a, b]$," se puede formalizar definiendo las fórmulas atómicas $p =$ la función f es derivable en $[a, b]$ y $q =$ la función f es continua en $[a, b]$ y aplicando el conectivo de implicación. Se obtiene $(p \to q)$ y, en forma abreviada, $p \to q$.
2) La frase "Si salto por la ventana, o me hago daño o empiezo a volar" se puede formalizar por medio de las proposiciones atómicas $p =$ salto por la ventana, $q =$ me hago daño, $r =$ empiezo a volar y los conectivos de implicación y de disyunción. Se obtiene $(p \to (q \vee r))$ y, en forma abreviada, $p \to (q \vee r)$.
3) La frase "Si salto por la ventana me podría hacer daño" se puede formalizar por medio de las proposiciones atómicas $p =$ salto por la ventana, $q =$ me podría hacer daño y el conectivo de implicación. Se obtiene $(p \to q)$ y, en forma abreviada, $p \to q$.

Formalización

Ejemplos

4) El enunciado "Condición necesaria y suficiente para que un número entero n *sea par es que* n *sea divisible por 2" se puede formalizar por medio de las proposiciones atómicas* $p =$ *n es un número entero,* $q =$ *n es par,* $r =$ *n es divisible por 2 y los conectivos de conjunción y doble implicación. Se obtiene* $((p \wedge q) \leftrightarrow (p \wedge r))$ *y, en forma abreviada,* $p \wedge q \leftrightarrow p \wedge r$.

5) Consideremos el razonamiento "Me gusta el helado de fresa, pero también el de limon. Si hay sólo helado de chocolate lo comeré, a pesar de que no me guste. Por tanto, no comeré helado de fresa." Para formalizar el razonamiento dado, definimos las proposiciones atómicas $p =$ *me gusta el helado de fresa,* $q =$ *me gusta el helado de limon,* $r =$ *hay sólo helado de chocolate,* $s =$ *comeré helado de chocolate,* $t =$ *me gusta el helado de chocolate,* $u =$ *comeré helado de fresa. La formalización se puede escribir, en forma abreviada, como:*

$$p \wedge q \wedge (r \rightarrow s) \wedge \neg t \rightarrow \neg u$$