

UNIVERSIDAD AUTÓNOMA DE MÉXICO

Facultad de Ciencias

Autor: Adrián Aguilera Moreno



Lógica Computacional

Semanal 3

Para cada uno de los siguientes ejercicios, **justifica ampliamente** tu respuesta.

1. **Realiza** las siguientes sustituciones **eliminando** los paréntesis superfluos en el resultado y **mostrando paso a paso** el procedimiento.

$$a) ((q \vee r)[q, p := \neg p, s] \rightarrow (r \wedge \neg(r \leftrightarrow p)))[p, r, q := r \vee q, q \wedge p, s].$$

$$b) (u \vee t) \rightarrow (\neg r \leftrightarrow (u \leftrightarrow s))[r, u, t := u, t, r].$$

▽ **Solución:** Analizando los incisos tenemos que

a) Sea $\alpha = [q, p := \neg p, s]$ y $\beta = [p, r, q := r \vee q, q \wedge p, s]$, entonces

$$\begin{aligned} ((q \vee r)\alpha \rightarrow (r \wedge \neg(r \leftrightarrow p)))\beta &= ((q\alpha \vee r\alpha) \rightarrow (r \wedge \neg(r \leftrightarrow p)))\beta \\ &= ((\neg p \vee r) \rightarrow (r \wedge \neg(r \leftrightarrow p)))\beta \\ &= ((\neg p \vee r)\beta \rightarrow (r \wedge \neg(r \leftrightarrow p))\beta) \\ &= \neg p\beta \vee r\beta \rightarrow r\beta \wedge \neg(r \leftrightarrow p)\beta \\ &= \neg(r \vee q) \vee (q \wedge p) \rightarrow (q \wedge p) \wedge \neg(r\beta \leftrightarrow p\beta) \\ &= \neg(r \vee q) \vee (q \wedge p) \rightarrow (q \wedge p) \wedge \neg(q \wedge p \leftrightarrow r \vee q) \end{aligned}$$

b) Sea $\alpha = [r, u, t := u, t, r]$, entonces

$$\begin{aligned} (u \vee t) \rightarrow (\neg r \leftrightarrow (u \leftrightarrow s))\alpha &= (u \vee t) \rightarrow (\neg r\alpha \leftrightarrow (u \leftrightarrow s)\alpha) \\ &= (u \vee t) \rightarrow (\neg u \leftrightarrow (u\alpha \leftrightarrow s\alpha)) \\ &= (u \vee t) \rightarrow (\neg u \leftrightarrow (t \leftrightarrow s)) \\ &= u \vee t \rightarrow (\neg u \leftrightarrow (t \leftrightarrow s)) \end{aligned}$$

◁

2. a) **Define recursivamente** la función pa que dada una fórmula φ , devuelve el número de paréntesis abiertos “(” que tiene φ .
 b) **Define recursivamente** la función pc que dada una fórmula φ , devuelve el número de paréntesis cerrados “)” que tiene φ .
 c) Sea $\varphi = (((\neg p \wedge q) \vee \neg r) \rightarrow r)$. Demuestra que

$$pa(\varphi) - pc(\varphi) = 0$$

▽ **Solución:**

a) Definimos $pa : PROP \rightarrow \mathbb{N} \cup \{0\}$. Así, para

$$\varphi \in ATOM$$

se tiene que $pa(\varphi) = 0$. Para $\alpha, \mu \in PROP$ se tiene que

$$\begin{aligned} pa((\alpha)) &= 1 + pa(\alpha) \\ pa(\neg\alpha) &= pa(\alpha) \\ pa(\alpha * \mu) &= pa(\alpha) + pa(\mu) \end{aligned}$$

donde $*$ $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

b) Definimos $pc : PROP \rightarrow \mathbb{N} \cup \{0\}$. Así, para

$$\varphi \in ATOM$$

se tiene que $pc(\varphi) = 0$. Para $\alpha, \mu \in PROP$ se tiene que

$$\begin{aligned} pc((\alpha)) &= pc(\alpha) + 1 \\ pc(\neg\alpha) &= pc(\alpha) \\ pc(\alpha * \mu) &= pc(\alpha) + pc(\mu) \end{aligned}$$

donde $*$ $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

c) Veamos que

$$\begin{aligned} pa((((\neg p \wedge q) \vee \neg r) \rightarrow r)) &= 1 + pa(((\neg p \wedge q) \vee \neg r) \rightarrow r) \\ &= 1 + pa(((\neg p \wedge q) \vee \neg r)) + pa(r) \\ &= 2 + pa((\neg p \wedge q)) + pa(\neg r) \\ &= 3 + pa(\neg p \wedge q) + pa(r) \\ &= 3 + pa(\neg p) + pa(q) \\ &= 3 + pa(p) \\ &= 3 \end{aligned}$$

además, se tiene que

$$\begin{aligned} pc((((\neg p \wedge q) \vee \neg r) \rightarrow r)) &= pc(((\neg p \wedge q) \vee \neg r) \rightarrow r) + 1 \\ &= pc(((\neg p \wedge q) \vee \neg r)) + pc(r) + 1 \\ &= pc((\neg p \wedge q)) + pc(\neg r) + 2 \\ &= pc(\neg p \wedge q) + pc(r) + 3 \\ &= pc(\neg p) + pc(q) + 3 \\ &= pc(p) + 3 \\ &= 3 \end{aligned}$$

Luego, es claro que

$$\begin{aligned} pa((((\neg p \wedge q) \vee \neg r) \rightarrow r)) - pc((((\neg p \wedge q) \vee \neg r) \rightarrow r)) &= 3 - 3 \\ &= 0 \end{aligned}$$

◁

Desafío extra...

- **Define recursivamente** una función `compress` que elimina los elementos consecutivos repetidos de una lista.

Ejemplo:

$$\text{compress}([a, a, a, a, i, i, i, u, d, d, d, d, d, a, a, a, a, a, a, a]) = [a, i, u, d, a]$$

▽ **Solución:** Para este inciso definimos una función auxiliar que nos indique si un elemento arbitrario esta contenido en una lista, esto es

$$\text{contains}(x:xs, a)^1 = \begin{cases} \text{true} & \text{si } x = a \\ \text{false} & \text{si } x = \phi \\ \text{contains}(s, a) & \text{si } x \neq a \end{cases}$$

¹`contains: [_] → Bool.`

Ahora definamos la función $\text{compress}:[_] \rightarrow [_]$, esto es

$$\text{compress}(x:xs) = \begin{cases} \text{compress}(s) & \text{si } \text{contains}(s, x) = \text{true} \\ x ++ \text{compress}(s) & \text{si } \text{contains}(s, x) = \text{false} \end{cases}$$

<

– **Demuestra**, usando tu definición, que:

$$\text{compress}([1, 2, 2, 3, 3, 3]) = [1, 2, 3]$$

Demostración: Veamos la ejecución de compress en las siguientes líneas:

$$\begin{aligned} \text{compress}([1, 2, 2, 3, 3, 3]) &= [1] ++ \text{compress}([2, 2, 3, 3, 3]) \\ &= [1] ++ \text{compress}([2, 3, 3, 3]) \\ &= [1] ++ [2] ++ \text{compress}([3, 3, 3]) \\ &= [1] ++ [2] ++ \text{compress}([3, 3]) \\ &= [1] ++ [2] ++ \text{compress}([3]) \\ &= [1] ++ [2] ++ [3] \\ &= [1, 2, 3] \end{aligned}$$

Cabe destacar que por cada llamada a $\text{compress}(x)$ se llama a lo más $\frac{|x| \cdot (|x|+1)}{2}$ veces a la función contains o al menos 1 vez. \square