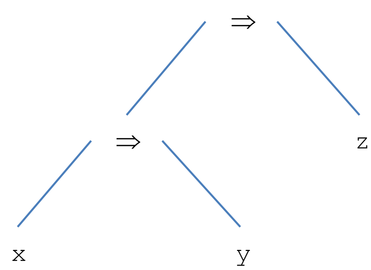
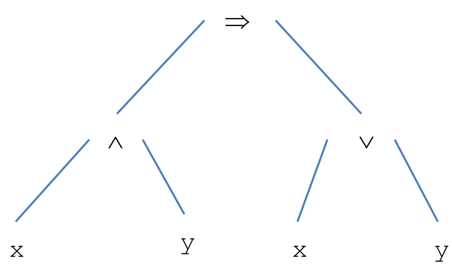
2.1.1

2

2a $(x \Rightarrow y) \Rightarrow z$ **:**



2b $x \wedge y \Rightarrow x \vee y$ **:**



2.1.2

1

1a Hay que expresar cada uno de los operadores de la tabla de 2.1.2.3 en términos de $\wedge$ y $\neg$ y las variables x, y. Nótese que se describen como operadores binarios, así sean esencialmente constantes o unarios.

$\mathbf{F}(x,y)$	$\equiv$	$x \wedge \neg x$
$x \wedge y$	$\equiv$	$x \wedge y$
$x \not\Rightarrow \mathbf{y}$	$\equiv$	$x \wedge \neg y$
$(x,y)_1$	$\equiv$	y
$\mathbf{x} \not\Leftarrow \mathbf{y}$	$\equiv$	$\neg x \wedge y$
$(x,y)_2$	$\equiv$	x
$x \not\equiv y$	$\equiv$	$\neg((x \wedge y) \wedge (\neg x \wedge \neg y))$
$x \vee y$	$\equiv$	$\neg(x \wedge y)$
x **nor** y	$\equiv$	$\neg x \wedge \neg y$
$x \equiv y$	$\equiv$	$\neg((\neg x \wedge y) \wedge (x \wedge \neg y))$
$(x,y)\neg_2$	$\equiv$	$\neg y$
$x \Leftarrow y$	$\equiv$	$\neg(\neg x \wedge y)$
$(x,y)\neg_1$	$\equiv$	$\neg x$
$x \Rightarrow y$	$\equiv$	$\neg(x \wedge \neg y)$
x **nand** y	$\equiv$	$\neg(x \wedge y)$
$\mathbf{T}(x,y)$	$\equiv$	$\neg(x \wedge \neg x)$

1b Por ejemplo, la constante **F** (siempre `false`) y la negación $\neg x$ no pueden expresarse en términos de disyunciones y conjunciones de las variables x, y. Para probar esta clase de afirmaciones se requiere una técnica de demostraciones más avanzada que la deducción que se ha explicado. Sin embargo, si se observan las fórmulas que se pueden construir, siempre las variables x e y aparecen afirmadas, y las conjunciones y disyunciones involucradas deben reducirse a x, y, $x \vee y$ o $x \wedge y$.

Los 3 argumentos representan 8 posibles estados de evaluación. Cada estado puede ser `true` o `false`, i.e., 2 valores por estado. Dos operadores son diferentes si evalúan diferente en un estado. Entonces, para contar los operadores ternarios diferentes hay que contar cuántas formas de evaluar hay, en los 8 estados. Estas son $2^8 = 256$.

2.1.3

1 $x \Rightarrow y \equiv \neg x \vee y$

Es tautología. La siguiente tabla de verdad lo comprueba:

x	**y**	**x ⇒ y**	**¬x**	**¬x ∨ y**	**x ⇒ y ≡ ¬x ∨ y**
0	**0**	1	1	1	**1**
0	**1**	1	1	1	**1**
1	**0**	0	0	0	**1**
1	**1**	1	0	1	**1**

2 $(x \Rightarrow y) \Rightarrow x$

No es tautología. Nótese que en el estado en que x = `false`, y = `false` se obtiene como valor de verdad `false`.
No es contradicción. Nótese que en el estado en que x = `true`, y = `true` se obtiene como valor de verdad `true`.

3 $x \Rightarrow y \equiv x \vee \neg y$

No es tautología. Nótese que en el estado en que x = `true`, y = `false` se obtiene como valor de verdad `false` al lado izquierdo de la equivalencia y `true` al lado derecho, i.e., evalúa a `false`.
No es contradicción. Nótese que en el estado en que x = `true`, y = `true` se obtiene como valor de verdad `true` en los dos lados, i.e., evalúa a `true`.

4 $x \wedge (x \Rightarrow y) \wedge \neg y$

Es contradicción. La siguiente tabla de verdad lo comprueba:

x	y	x ⇒ y	¬y	x ∧ (x ⇒ y)	x ∧ (x ⇒ y) ∧ ¬y
0	0	1	1	0	0
0	1	1	0	0	0
1	0	0	1	0	0
1	1	1	0	1	0

2.2.1

1
1a
Variables proposicionales:
`ma` **:** hoy es martes

mi **:** hoy es miércoles
Traducción:
$ma \Rightarrow \neg mi$
Contrarrecíproca
$mi \Rightarrow \neg ma$
Si hoy es miércoles, entonces hoy no es martes.

1b
Variables proposicionales:
l **:** llueve
m **:** voy a hacer mercado
Traducción:
$l \Rightarrow \neg m$
Contrarrecíproca
$m \Rightarrow \neg l$
Si hoy voy a hacer mercado, entonces no llueve.

1c
Variables proposicionales:
s **:** me quedaré solo
v **:** te vas
Traducción:
$v \Rightarrow s$
Contrarrecíproca
$\neg s \Rightarrow \neg v$
Si no me quedo solo, entonces tú no te vas.

1d
Variables proposicionales:
q **:** me quedaré
v **:** te vas
Traducción:
$q \equiv v$
Contrarrecíproca
$\neg q \equiv \neg v$
No me quedaré solo si tú no te vas.

Otra posibilidad (dependiendo de la intención que se quiera transmitir):
Traducción:
$v \Rightarrow q$
Contrarrecíproca
$\neg q \Rightarrow \neg v$
No me quedaré, luego tú no te vas.

1e
Variables proposicionales:
tt **:** puedo terminar la tarea
ed **:** entiendo la demostración
Traducción:
$\neg tt \Rightarrow \neg ed$
Contrarrecíproca
$ed \equiv tt$
Si entiendo la demostración puedo terminar la tarea.

1f Variables proposicionales:

l **:** está lloviendo

p **:** puedo ir al pueblo

Traducción:

$l \Rightarrow \neg p$

Contrarrecíproca

$p \Rightarrow \neg l$

Si puedo ir al pueblo, no llueve.

3

a *Definición de variables*

E1 Si llueve Juan se queda en casa.

l: Llueve

c: Juan se queda en casa

E2 Si Juan tiene tareas se queda en casa.

t: Juan tiene tareas

E3 Juan se quedó en casa.

E4 Juan tiene tareas o está lloviendo.

b *Traducción a lógica proposicional*

(E1) $l \Rightarrow c$

(E2) $t \Rightarrow c$

(E3) c

(E4) $t \vee l$

La expresión correspondiente es

$$E1 \wedge E2 \wedge E3 \Rightarrow E4$$

c *Tabla de verdad*

l	**c**	**t**	**l ⇒ c**	**t ⇒ c**	**l ∨ t**	**E1 ∧ E2 ∧ E3**	**E1 ∧ E2 ∧ E3 ⇒ E4**
0	0	0	1	1	0	0	1
0	0	1	1	0	1	0	1
0	1	0	1	1	0	1	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1
1	0	1	0	0	1	0	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

El argumento no es correcto en el estado en que $l \equiv false$, $c \equiv true$ y $t \equiv false$, i.e., cuando no llueve, Juan se queda en casa y no tiene tareas. En este estado las tres premisas son verdaderas, pero la conclusión no. Esencialmente, la incorrección del razonamiento estaría en creer que si $p \Rightarrow q$ y se diera q, entonces debiera darse p.

2.4.1

1a $p \wedge (q \Rightarrow p) \equiv \neg p$

Cuando $p \equiv true$, el lado izquierdo de la equivalencia es $true$, pero el lado derecho vale $false$. En otras palabras, no es una fórmula válida y, por tanto, no puede ser un axioma.

1b $(p \Rightarrow q) \wedge (q \Rightarrow r) \wedge (r \Rightarrow s) \equiv p \Rightarrow s$

La siguiente tabla de verdad comprueba que la expresión es una tautología, de manera que sí puede ser un axioma.

				[1]	[2]	[3]	[4]	[5]	
p	q	r	s	$p \Rightarrow q$	$q \Rightarrow r$	$r \Rightarrow s$	$[1] \wedge [2] \wedge [3]$	$p \Rightarrow s$	[4] ≡]5]
0	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1
0	0	1	0	1	1	0	0	1	1
0	0	1	1	1	1	1	1	1	1
0	1	0	0	1	0	1	0	1	1
0	1	0	1	1	0	1	0	1	1
0	1	1	0	1	1	0	0	1	1
0	1	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	0	1
1	0	0	1	0	1	1	0	0	1
1	0	1	0	0	1	0	0	0	1
1	0	1	1	0	1	1	0	0	1
1	1	0	0	1	0	1	0	1	1
1	1	0	1	1	0	1	0	1	1
1	1	1	0	1	1	0	0	1	1
1	1	1	1	1	1	1	1	1	1

2 En una tabla copia de la de 2.1.2.3 se puede buscar qué operadores cumplen los axiomas correspondientes. A continuación se ilustra esta idea, llamando • a un operador que cumple el axioma mencionado. Las columnas en rojo descartan operadores que no cumplen el axioma dado:

•-Identidad

x • false ≡ x

x	y																
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

•-Conmutatividad

x • y ≡ y • x

x	y																
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

•-Dominancia

X • true ≡ true

x	y																
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

La única columna que no es roja en ninguna de los tres casos (porque cumple los axiomas) es la que tiene la forma

x	y	
0	0	0
0	1	1
1	0	1
1	1	1

y esta es, precisamente, la que define el operador ∨.

2.4.2

1

1a	`x[x:= a+3] = a+3`
1b	`x+y*x[x:= a+3] = x+y*(a+3)`
1c	`(x+y*x)[x:= a+3] = a+3+y*(a+3)`
1d	`x+y*x[x,y:= y,x] = x+x*y`
1e	`(x+y*x)[x,y:= y,x] = y+x*y`
1f	`(b[i]+5*b[j])[i,b[i],b[j]:= i+1,2,3] = 2+5*3   // si i≠j`
1g	`(b[i]+5*b[j])[i:= i+1][b[i]:= 2][b[j]:= 3] = b[i+1]+5*3`

2 La regla de Leibniz se puede escribir con ayuda de la notación de sustitución así;

$$\textit{Leibniz}:\quad \frac{\texttt{p} \equiv \texttt{q}}{\texttt{E[z:= p]} \equiv \texttt{E[z:= q]}} \qquad \begin{array}{|l} \texttt{p,q}: \text{ proposiciones} \\ \texttt{E[x]}: \text{ expresión sobre } \texttt{x} \end{array}$$

Basta mostrar que, cuando se dan las hipótesis de una de las dos regla s, se puede deducir su conclusión con ayuda de la otra regla.
Si en un momento valen `Z≡X` y `Z≡Y`, la transitividad de ≡ permite afirmar que `X≡Y`. Si se tiene la Regla de Leibniz, se puede concluir que `E[z:= X] ≡ E[z:= Y]`. Esta es la conclusión que se tendría al aplicar Leibniz1 a las hipótesis iniciales.
Si en un momento vale `X ≡ Y`, la reflexividad de ≡ permite afirmar que `X ≡ Y`, `Y ≡ Y`. Si se tiene la Regla de Leibniz1, se puede concluir que `E[z:= X] ≡ E[z:= Y]`. Esta es la conclusión que se tendría al aplicar Leibniz a las hipótesis iniciales.

2.4.3

1 **Teo:** `x ∧ y ≡ x ∨ y ≡ x ≡ y`

Dem:

```
      x ∧ y ≡ x ∨ y ≡ x ≡ y
=           〈 Def ≡ 〉
      x ∧ y ≡ x ∨ y ≡ (x ∧ y) ∨ (¬x ∧ ¬y)
=           〈 ≡-Conm 〉
      x ∨ y ≡ x ∧ y ≡ (x ∧ y) ∨ (¬x ∧ ¬y)
=           〈 ∨-Ident 〉
      x ∨ y ≡ (x ∧ y) ∨ false ≡ (x ∧ y) ∨ (¬x ∧ ¬y)
=           〈 Distr ∨/≡ 〉
      x ∨ y ≡ (x ∧ y) ∨ (false ≡ (¬x ∧ ¬y))
=           〈 Def false 〉
      x ∨ y ≡ (x ∧ y) ∨(¬true ≡ (¬x ∧ ¬y))
=           〈 Neg ≡ 〉
      x ∨ y ≡ (x ∧ y) ∨ ¬(true ≡ (¬x ∧ ¬y))
=           〈 ≡-Ident 〉
      x ∨ y ≡ (x ∧ y) ∨ ¬(¬x ∧ ¬y)
=           〈 De Morgan 〉
      x ∨ y ≡ (x ∧ y) ∨ x ∨ y
=           〈 ∨-Conm 〉
```

$x \vee y \equiv x \vee (x \wedge y) \vee y$

= ⟨ Absorción ⟩

$x \vee y \equiv x \vee y$

= ⟨ ≡-Identidad ⟩

true

[]

2 Teo: $(x \Rightarrow y) \wedge \neg(x \equiv y) \Rightarrow y$

Dem:

$(x \Rightarrow y) \wedge \neg(x \equiv y) \Rightarrow y$

= ⟨ Def ⇒ ⟩

$(\neg x \vee y) \wedge \neg(x \equiv y) \Rightarrow y$

= ⟨ Def ⇒⟩

$\neg((\neg x \vee y) \wedge \neg(x \equiv y)) \vee y$

= ⟨ De Morgan ⟩

$(\neg(\neg x \vee y) \vee \neg\neg(x \equiv y)) \vee y$

= ⟨ De Morgan ⟩

$((\neg\neg x \wedge \neg y) \vee \neg\neg(x \equiv y)) \vee y$

= ⟨ Doble negación, 2 veces ⟩

$((x \wedge \neg y) \vee (x \equiv y)) \vee y$

= ⟨ ∨-Asoc, ∨-Conm ⟩

$(x \wedge \neg y) \vee y \vee (x \equiv y)$

= ⟨ Absorción ¬ ⟩

$(x \vee y) \vee (x \equiv y)$

= ⟨ Distribución ∨/≡⟩

$x \vee y \vee x \equiv x \vee y \vee y$

= ⟨ ∨-Asoc, ∨-Conm ⟩

$x \vee x \vee y \equiv x \vee y \vee y$

= ⟨∨-Idempot, 2 veces⟩

$x \vee y \equiv x \vee y$

= ⟨∨-Conm. ≡-identidad⟩

true

[]

3 Teo: $p \wedge (p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow r$

Dem:

$p \wedge (p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow r$

= ⟨ Def ⇒ 2 veces ⟩

$p \wedge (\neg p \vee q) \wedge (\neg q \vee r) \Rightarrow r$

= ⟨Def ⇒ ⟩

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

= ⟨ Absorción ¬ ⟩

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

= ⟨ Absorción ¬ ⟩

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

= ⟨ De Morgan ⟩

$\neg p \vee \neg q \vee \neg r \vee r$

= ⟨ Medio excluido ⟩

$\neg p \vee \neg q \vee true$

= ⟨∨-Dominancia⟩

true

[]

4 Teo: $p \vee \neg p \equiv ((p \vee q) \wedge \neg(\neg p \wedge (\neg q \vee \neg r))) \vee (\neg p \wedge \neg q) \vee (\neg p \wedge \neg r)$

Dem:

$p \vee \neg p \equiv ((p \vee q) \wedge \neg(\neg p \wedge (\neg q \vee \neg r))) \vee (\neg p \wedge \neg q) \vee (\neg p \wedge \neg r)$

= ⟨ Medio excluido ⟩

$true \equiv ((p \vee q) \wedge \neg(\neg p \wedge (\neg q \vee \neg r))) \vee (\neg p \wedge \neg q) \vee (\neg p \wedge \neg r)$

= ⟨ ≡-Ident ⟩

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

= ⟨Distr ∧/∨⟩

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

= ⟨De Morgan, 2 veces ⟩

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

= ⟨De Morgan, dos veces ⟩

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

= ⟨Doble neg, 2 veces ⟩

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

= ⟨ Distr ∨/∧⟩

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

= ⟨∧-Asoc ⟩

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

= ⟨∧-Idempot ⟩

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

≡ ⟨ Medio excluido ⟩

true

[]

5 Teo: $(p \vee q) \wedge (\neg p \wedge (\neg p \wedge q)) \equiv (\neg p \wedge q)$

Dem:

$(p \vee q) \wedge (\neg p \wedge (\neg p \wedge q))$

= ⟨ ∧-Asoc ⟩

$(p \vee q) \wedge (\neg p \wedge \neg p \wedge q)$

= ⟨ ∧-idemp ⟩

$(p \vee q) \wedge (\neg p \wedge q)$

= ⟨ Distr ∧/∨⟩

$(p \wedge \neg p \wedge q) \vee (q \wedge \neg p \wedge q)$

= ⟨ Contradicción ⟩

$(false \wedge q) \vee (q \wedge \neg p \wedge q)$

= ⟨∧-Dominancia⟩

$false \vee (q \wedge \neg p \wedge q)$

= ⟨∨-Ident ⟩

$q \wedge \neg p \wedge q$

= ⟨∧-Conm ⟩

$\neg p \wedge q \wedge q$

= ⟨∧-Idemp ⟩

$\neg p \wedge q$

[]

6 Teo: $p \Rightarrow (q \Rightarrow p) \equiv \neg p \Rightarrow (p \Rightarrow q)$

Dem:

$p \Rightarrow (q \Rightarrow p) \equiv \neg p \Rightarrow (p \Rightarrow q)$

= ⟨Def ⇒, 2 veces ⟩

$p \Rightarrow (\neg q \vee p) \equiv \neg p \Rightarrow (\neg p \vee q)$

$=$ ⟨Def ⇒, 2 veces ⟩
¬p ∨ (¬q ∨ p) ≡ ¬¬p ∨ (¬p ∨ q)
$=$ ⟨ Doble neg ⟩
¬p ∨ ¬q ∨ p ≡ p ∨ ¬p ∨ q
$=$ ⟨∨-Conm ⟩
¬p ∨ p ∨ ¬q ≡ p ∨ ¬p ∨ q
$=$ ⟨ Medio excluido, dos veces ⟩
true ∨ ¬q ≡ true ∨ q
$=$ ⟨∨-Dominancia, 2 veces ⟩
true ≡ true
$=$ ⟨≡-Ident ⟩
true

[]

7 **Teo:** (p ⇒ q) ∧ (r ⇒ q) ≡ (p ∨ r) ⇒ q
Dem:
(p ⇒ q) ∧ (r ⇒ q) ≡ (p ∨ r) ⇒ q
$=$ ⟨Def ⇒, 3 veces ⟩
(¬p ∨ q) ∧ (¬r ∨ q) ≡ ¬(p ∨ r) ∨ q
$=$ ⟨ Distr ∨/∧⟩
(¬p ∧ ¬r) ∨ q ≡ ¬(p ∨ r) ∨ q
$=$ ⟨ De Morgan ⟩
¬(p ∨ r) ∨ q ≡ ¬(p ∨ r) ∨ q
$=$ ⟨≡-Ident ⟩
true

[]

2.4.4

1 **Teo:** p ∧ q ∧ r ⇒ p
Dem: Prueba por hipótesis
Hip: p, q, r // A demostrar: p
p
$=$ ⟨ Hip: p ⟩
true

[]

2 **Teo:** p ∧ ¬(¬r ∨ ¬p) ⇒ r ∧ p
Dem: Prueba por hipótesis
Hip: p, ¬(¬r ∨ ¬p) // A demostrar: r ∧ p
true
$=$ ⟨ Hip: ¬(¬r ∨ ¬p) ⟩
¬(¬r ∨ ¬p)
$=$ ⟨ De Morgan ⟩
¬¬r ∧ ¬¬p
$=$ ⟨ Doble neg, 2 veces ⟩
r ∧ p

[]

3 **Teo:** (x ∨ y) ∧ (x ≡ y) ⇒ (x ⇒ y)
Dem: Prueba por hipótesis
Hip: x ∨ y, x ≡ y // A demostrar: x ⇒ y

4 **Teo**: (p ∨ q) ∧ (p ≡ q) ⇒ p ∧ q
Dem: Prueba por hipótesis
Hip: p ∨ q, p ≡ q // A demostrar: p ∧ q
Casos: p, q

```
        p ∨ q
=             ⟨ Hip: p ∨ q ⟩
        true
```

Caso p:

```
        p ∧ q
=             ⟨ Hip: p ≡ q ⟩
        p ∧ p
=             ⟨ ∧-Idemp ⟩
        p
=             ⟨ Caso: p ⟩
        true
```

Caso q:

```
        p ∧ q
=             ⟨ Hip: p ≡ q ⟩
        q ∧ q
=             ⟨ ∧-Idemp ⟩
        q
=             ⟨ Caso: q ⟩
        true
```

[]

5 **Teo**: p ∧ (p ⇒ q) ∧ (q ⇒ r) ⇒ r
Dem: Prueba por hipótesis
Hip: p, p ⇒ q, q ⇒ r // A demostrar: r

```
              ⟨ Hip: p ⟩
        p
⇒             ⟨ Hip: p ⇒ q¸ Modus Ponens ⟩
        q
⇒             ⟨ Hip: q ⇒ r¸ Modus Ponens ⟩
        r
```

[]

6 (p ⇒ q) ∧ (r ⇒ s) ⇒ (p ∨ r ⇒ q ∨ s)
Dem: Prueba por hipótesis
Hip: p ⇒ q, r ⇒ s // A demostrar: p ∨ r ⇒ q ∨ s

Prueba por hipótesis
Hip: p ∨ r // A demostrar: q ∨ s

Casos: p ∨ r

```
        p ∨ r
=             ⟨ Hip: p ∨ r ⟩
        true
```

Caso p:

```
              ⟨ Caso: p ⟩
        p
⇒             ⟨ Hip: p ⇒ q¸ Modus Ponens ⟩
```

$$
\begin{array}{lll}
 & q & \\
\Rightarrow & & \langle \text{ Debilitamiento } \rangle \\
 & q \vee s &
\end{array}
$$

Caso r:

$$
\begin{array}{lll}
 & & \langle \text{ Caso: } r \rangle \\
 & r & \\
\Rightarrow & & \langle \text{ Hip: } r \Rightarrow s\text{, Modus Ponens } \rangle \\
 & s & \\
\Rightarrow & & \langle \text{ Debilitamiento } \rangle \\
 & q \vee s &
\end{array}
$$

$\square$