

# Ejercicios de lógica proposicional con Lean

José A. Alonso Jiménez

---

Grupo de Lógica Computacional  
Dpto. de Ciencias de la Computación e Inteligencia Artificial  
Universidad de Sevilla  
Sevilla, 9 de diciembre de 2020

Esta obra está bajo una licencia Reconocimiento-NoComercial-CompartirIgual 2.5 Spain de Creative Commons.

**Se permite:**

- copiar, distribuir y comunicar públicamente la obra
- hacer obras derivadas

**Bajo las condiciones siguientes:**

**Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor.



**No comercial.** No puede utilizar esta obra para fines comerciales.



**Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- Algunas de estas condiciones pueden no aplicarse si se obtiene el permiso del titular de los derechos de autor.

Esto es un resumen del texto legal (la licencia completa). Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/es/> o envíe una carta a Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

# Índice general

<b>1 Introducción</b>	<b>5</b>
<b>2 Ejercicios sobre condicionales</b>	<b>7</b>
2.1 $p \rightarrow q, p \vdash q$	7
2.2 $p \rightarrow q, q \rightarrow r, p \vdash r$	8
2.3 $p \rightarrow (q \rightarrow r), p \rightarrow q, p \vdash r$	9
2.4 $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$	11
2.5 $p \rightarrow (q \rightarrow r) \vdash q \rightarrow (p \rightarrow r)$	13
2.6 $p \rightarrow (q \rightarrow r) \vdash (p \rightarrow q) \rightarrow (p \rightarrow r)$	16
2.7 $p \vdash q \rightarrow p$	18
2.8 $\vdash p \rightarrow (q \rightarrow p)$	20
2.9 $p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)$	21
2.10 $p \rightarrow (q \rightarrow (r \rightarrow s)) \vdash r \rightarrow (q \rightarrow (p \rightarrow s))$	23
2.11 $\vdash (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$	25
2.12 $(p \rightarrow q) \rightarrow r \vdash p \rightarrow (q \rightarrow r)$	28
<b>3 Ejercicios sobre conjunciones</b>	<b>31</b>
3.1 $p, q \vdash p \wedge q$	31
3.2 $p \wedge q \vdash p$	32
3.3 $p \wedge q \vdash q$	34
3.4 $p \wedge (q \wedge r) \vdash (p \wedge q) \wedge r$	35
3.5 $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$	38
3.6 $p \wedge q \vdash p \rightarrow q$	39
3.7 $(p \rightarrow q) \wedge (p \rightarrow r) \vdash p \rightarrow q \wedge r$	40
3.8 $p \rightarrow q \wedge r \vdash (p \rightarrow q) \wedge (p \rightarrow r)$	43
3.9 $p \rightarrow (q \rightarrow r) \vdash p \wedge q \rightarrow r$	45
3.10 $p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$	47
3.11 $(p \rightarrow q) \rightarrow r \vdash p \wedge q \rightarrow r$	48
3.12 $p \wedge (q \rightarrow r) \vdash (p \rightarrow q) \rightarrow r$	50

<b>4 Ejercicios sobre disyunciones</b>	<b>53</b>
4.1 $p \vdash p \vee q$	53
4.2 $q \vdash p \vee q$	54
4.3 $p \vee q \vdash q \vee p$	55
4.4 $q \rightarrow r \vdash p \vee q \rightarrow p \vee r$	57
4.5 $p \vee p \vdash p$	60
4.6 $p \vdash p \vee p$	62
4.7 $p \vee (q \vee r) \vdash (p \vee q) \vee r$	62
4.8 $(p \vee q) \vee r \vdash p \vee (q \vee r)$	66
4.9 $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$	68
4.10 $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$	70
4.11 $p \vee (q \wedge r) \vdash (p \vee q) \wedge (p \vee r)$	72
4.12 $(p \vee q) \wedge (p \vee r) \vdash p \vee (q \wedge r)$	78
4.13 $(p \rightarrow r) \wedge (q \rightarrow r) \vdash p \vee q \rightarrow r$	81
4.14 $p \vee q \rightarrow r \vdash (p \rightarrow r) \wedge (q \rightarrow r)$	84
<b>5 Ejercicios sobre negaciones</b>	<b>89</b>
5.1 $p \vdash \neg\neg p$	89
5.2 $\neg p \vdash p \rightarrow q$	90
5.3 $p \rightarrow q \vdash \neg q \rightarrow \neg p$	92
5.4 $p \vee q, \neg q \vdash p$	94
5.5 $p \vee q, \neg p \vdash q$	96
5.6 $p \vee q \vdash \neg(\neg p \wedge \neg q)$	98
5.7 $p \wedge q \vdash \neg(\neg p \vee \neg q)$	100
5.8 $\neg(p \vee q) \vdash \neg p \wedge \neg q$	101
5.9 $\neg p \wedge \neg q \vdash \neg(p \vee q)$	103
5.10 $\neg p \vee \neg q \vdash \neg(p \wedge q)$	105
5.11 $\neg(p \wedge \neg p)$	107
5.12 $p \wedge \neg p \vdash q$	108
<b>6 Ejercicios de lógica clásica</b>	<b>111</b>
6.1 $\neg\neg p \vdash p$	111
6.2 $\vdash p \vee \neg p$	112
6.3 $((p \rightarrow q) \rightarrow p) \rightarrow p$	116
6.4 $\neg q \rightarrow \neg p \vdash p \rightarrow q$	118
6.5 $\neg(\neg p \wedge \neg q) \vdash p \vee q$	119
6.6 $\neg(\neg p \vee \neg q) \vdash p \wedge q$	121
6.7 $\neg(p \wedge q) \vdash \neg p \vee \neg q$	123
6.8 $\vdash (p \rightarrow q) \vee (q \rightarrow p)$	125
<b>7 Bibliografía</b>	<b>127</b>

# Capítulo 1

## Introducción

Este libro es una colección de soluciones de ejercicios de lógica proposicional formalizadas con [Lean](#) que complementa el libro de [Lógica con Lean](#).

En cada capítulo va entrando en juego una nueva conectiva y para cada uno de los ejercicios se formalizan las soluciones en distintos estilos:

- aplicativo usando tácticas con razonamiento hacia atrás,
- declarativo (o estructurado) con razonamiento hacia adelante,
- funcional con términos del tipo especificado y
- automático.

Las demostraciones funcionales se obtienen mediante una sucesión de transformaciones de una applicativa (o declarativa) eliminando elementos no esenciales.

Además, al final de cada ejercicio se encuentra un enlace al código y otro a una sesión de Lean en la Web que contiene la solución del ejercicio.



# Capítulo 2

## Ejercicios sobre condicionales

### 2.1. $p \rightarrow q, p \vdash q$

```
-- Ejercicio. Demostrar  
--       $p \rightarrow q, p \vdash q$   
-----
```

```
import tactic  
variables (p q : Prop)
```

```
-- 1ª demostración
```

```
example  
  (Hpq : p → q)  
  (Hp   : p)  
  : q :=  
Hpq Hp
```

```
-- 2ª demostración
```

```
example  
  (Hpq : p → q)  
  (Hp   : p)  
  : q :=  
by tauto
```

Enlaces al [código](#) y a la [sesión en Lean Web](#). y al vídeo.

## 2.2. $p \rightarrow q, q \rightarrow r, p \vdash r$

```
-- Ejercicio. Demostrar
--    $p \rightarrow q, q \rightarrow r, p \vdash r$ 
```

```
import tactic
variables (p q r : Prop)
```

```
-- 1ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  (Hp : p)
  : r :=
```

```
begin
  apply Hqr,
  apply Hpq,
  exact Hp,
```

```
end
```

```
-- 2ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  (Hp : p)
  : r :=
```

```
begin
  apply Hqr,
  exact Hpq Hp,
```

```
end
```

```
-- 3ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  (Hp : p)
  : r :=
```

```
begin
  exact Hqr (Hpq Hp),
```

```
end
```

```
-- 3ª demostración
```



```

example
  (Hpq : p → q)
  (Hqr : q → r)
  (Hp : p)
  : r :=
Hqr (Hpq Hp)

-- 4ª demostración
example
  (Hpq : p → q)
  (Hqr : q → r)
  (Hp : p)
  : r :=
by tauto

-- 5ª demostración
example
  (Hpq : p → q)
  (Hqr : q → r)
  (Hp : p)
  : r :=
have Hq : q,
  from Hpq Hp,
show r,
  from Hqr Hq

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.3. $p \rightarrow (q \rightarrow r), p \rightarrow q, p \vdash r$

```

-----
-- Ejercicio. Demostrar
--    $p \rightarrow (q \rightarrow r), p \rightarrow q, p \vdash r$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (Hpqr : p → (q → r))
  (Hpq : p → q)
  (Hp : p)

```

```

: r :=
begin
  have Hqr : q → r, from Hpqr Hp,
  apply Hqr,
  apply Hpq,
  exact Hp,
end

```

-- 2ª demostración

```

example
  (Hpqr : p → (q → r))
  (Hpq   : p → q)
  (Hp    : p)
  : r :=
begin
  have Hqr : q → r, from Hpqr Hp,
  apply Hqr,
  exact Hpq Hp,
end

```

-- 3ª demostración

```

example
  (Hpqr : p → (q → r))
  (Hpq   : p → q)
  (Hp    : p)
  : r :=
begin
  have Hqr : q → r, from Hpqr Hp,
  exact Hqr (Hpq Hp),
end

```

-- 4ª demostración

```

example
  (Hpqr : p → (q → r))
  (Hpq   : p → q)
  (Hp    : p)
  : r :=
  (Hpqr Hp) (Hpq Hp)

```

-- 5ª demostración

```

example
  (Hpqr : p → (q → r))
  (Hpq   : p → q)
  (Hp    : p)
  : r :=

```

```
-- by hint
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.4. $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$

```
-- Ejercicio. Demostrar
--    $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$ 
```

```
import tactic
variables (p q r : Prop)
```

```
-- 1ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
```

```
begin
  intro Hp,
  apply Hqr,
  apply Hpq,
  exact Hp,
end
```

```
-- 2ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
```

```
begin
  intro Hp,
  apply Hqr,
  exact Hpq Hp,
end
```

```
-- 3ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
```

```
begin
  intro Hp,
  exact Hqr (Hpq Hp),
end
```

```
-- 4ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
λ Hp, Hqr (Hpq Hp)
```

```
-- 5ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
assume Hp : p,
have Hq : q,
  from Hpq Hp,
show r,
  from Hqr Hq
```

```
-- 6ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
assume Hp : p,
have Hq : q,
  from Hpq Hp,
Hqr Hq
```

```
-- 7ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
assume Hp : p,
Hqr (Hpq Hp)
```

```
-- 8ª demostración
```

```
example
  (Hpq : p → q)
  (Hqr : q → r)
```

```

: p → r :=
λ Hp, Hqr (Hpq Hp)

-- 9ª demostración
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
-- by hint
by tauto

-- 10ª demostración
example
  (Hpq : p → q)
  (Hqr : q → r)
  : p → r :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.5. $p \rightarrow (q \rightarrow r) \vdash q \rightarrow (p \rightarrow r)$

```

-- -----
-- Ejercicio. Demostrar
--    $p \rightarrow (q \rightarrow r) \vdash q \rightarrow (p \rightarrow r)$ 
-- -----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
begin
  intro Hq,
  intro Hp,
  have Hqr : q → r,
    from Hpqr Hp,
  apply Hqr,
  exact Hq,
end

```

```

-- 2ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
begin
  intro Hq,
  intro Hp,
  have Hqr : q → r,
    from Hpqr Hp,
  exact Hqr Hq,
end

-- 3ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
begin
  intro Hq,
  intro Hp,
  exact (Hpqr Hp) Hq,
end

-- 4ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
begin
  intros Hq Hp,
  exact (Hpqr Hp) Hq,
end

-- 5ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
λ Hq Hp, (Hpqr Hp) Hq

-- 6ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
begin
  intros Hq Hp,
  apply Hpqr,
  { exact Hp, },

```

```
{ exact Hq, },
end

-- 7ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
assume Hq : q,
assume Hp : p,
have Hqr : q → r,
  from Hpqr Hp,
show r,
  from Hqr Hq

-- 8ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
assume Hq : q,
assume Hp : p,
have Hqr : q → r,
  from Hpqr Hp,
Hqr Hq

-- 9ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
assume Hq : q,
assume Hp : p,
(Hpqr Hp) Hq

-- 10ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
λ Hq Hp, (Hpqr Hp) Hq

-- 11ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
-- by hint
by tauto
```

```
-- 12ª demostración
example
  (Hpqr : p → (q → r))
  : q → (p → r) :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.6. $p \rightarrow (q \rightarrow r) \vdash (p \rightarrow q) \rightarrow (p \rightarrow r)$

```
-- Ejercicio. Demostrar
--   p → (q → r) ⊢ (p → q) → (p → r)
-- -----
```

```
import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
begin
  intros Hpq Hp,
  have Hqr : q → r,
    from Hpqr Hp,
  apply Hqr,
  apply Hpq,
  exact Hp,
end

-- 2ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
begin
  intros Hpq Hp,
  have Hqr : q → r,
    from Hpqr Hp,
  apply Hqr,
  exact Hpq Hp,
end
```



```

-- 3ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
begin
  intros Hpq Hp,
  have Hqr : q → r,
    from Hpqr Hp,
  exact Hqr (Hpq Hp),
end

-- 4ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
begin
  intros Hpq Hp,
  exact (Hpqr Hp) (Hpq Hp),
end

-- 5ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
λ Hpq Hp, (Hpqr Hp) (Hpq Hp)

-- 6ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,
have Hqr : q → r,
  from Hpqr Hp,
have Hq : q,
  from Hpq Hp,
show r,
  from Hqr Hq

-- 7ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,

```

```

have Hqr : q → r,
  from Hpqr Hp,
have Hq : q,
  from Hpq Hp,
Hqr Hq

-- 8ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,
have Hqr : q → r,
  from Hpqr Hp,
Hqr (Hpq Hp)

-- 9ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,
(Hpqr Hp) (Hpq Hp)

-- 10ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
λ Hpq Hp, (Hpqr Hp) (Hpq Hp)

-- 11ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
-- by hint
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.7. $p \vdash q \rightarrow p$

```
-----  
-- Ejercicio. Demostrar  
--    $p \vdash q \rightarrow p$   
-----  
  
import tactic  
variables (p q : Prop)  
  
-- 1ª demostración  
example  
  (Hp : p)  
  :  $q \rightarrow p$  :=  
begin  
  intro Hq,  
  exact Hp,  
end  
  
-- 2ª demostración  
example  
  (H : p)  
  :  $q \rightarrow p$  :=  
 $\lambda$  _, H  
  
-- 3ª demostración  
example  
  (Hp : p)  
  :  $q \rightarrow p$  :=  
assume Hq : q,  
show p,  
  from Hp  
  
-- 4ª demostración  
example  
  (Hp : p)  
  :  $q \rightarrow p$  :=  
-- by library_search  
imp_intro Hp  
  
-- 5ª demostración  
example  
  (Hp : p)  
  :  $q \rightarrow p$  :=  
-- by hint  
by tauto
```

```
-- 6ª demostración
example
  (Hp : p)
  : q → p :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.8. $\vdash p \rightarrow (q \rightarrow p)$

```
-- Ejercicio. Demostrar
--    $\vdash p \rightarrow (q \rightarrow p)$ 
--
```

```
import tactic
variables (p q : Prop)
```

```
-- 1ª demostración
example :
  p → (q → p) :=
begin
  intros Hp Hq,
  exact Hp,
end
```

```
-- 2ª demostración
example :
  p → (q → p) :=
λ Hp _, Hp
```

```
-- 3ª demostración
example :
  p → (q → p) :=
assume Hp : p,
assume Hq : q,
show p,
from Hp
```

```
-- 4ª demostración
example :
  p → (q → p) :=
-- by library_search
```

```

imp_intro

-- 5ª demostración
example :
  p → (q → p) :=
-- by hint
by tauto

-- 6ª demostración
example :
  p → (q → p) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.9. $p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)$

```

-----
-- Ejercicio. Demostrar
--    $p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
begin
  intros Hqr Hp,
  apply Hqr,
  apply Hpq,
  exact Hp,
end

-- 2ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
begin
  intros Hqr Hp,
  apply Hqr,

```

```

    exact Hpq Hp,
end

-- 3ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
begin
  intros Hqr Hp,
  exact Hqr (Hpq Hp),
end

-- 4ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
λ Hqr Hp, Hqr (Hpq Hp)

-- 5ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
assume Hqr : q → r,
assume Hp : p,
have Hq : q,
  from Hpq Hp,
show r,
  from Hqr Hq

-- 6ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
assume Hqr : q → r,
assume Hp : p,
have Hq : q,
  from Hpq Hp,
Hqr Hq

-- 7ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
λ Hqr Hp, Hqr (Hpq Hp)

```

```

-- 8ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
-- by hint
by tauto

-- 9ª demostración
example
  (Hpq : p → q)
  : (q → r) → (p → r) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.10. $p \rightarrow (q \rightarrow (r \rightarrow s)) \vdash r \rightarrow (q \rightarrow (p \rightarrow s))$

```

-----
-- Ejercicio. Demostrar
--    $p \rightarrow (q \rightarrow (r \rightarrow s)) \vdash r \rightarrow (q \rightarrow (p \rightarrow s))$ 
-----

import tactic
variables (p q r s : Prop)

-- 1ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
begin
  intros Hr Hq Hp,
  apply H,
  { exact Hp, },
  { exact Hq, },
  { exact Hr, },
end

-- 2ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
begin
  intros Hr Hq Hp,

```

```

exact H Hp Hq Hr,
end

-- 3ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
λ Hr Hq Hp, H Hp Hq Hr

-- 4ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
assume Hr : r,
assume Hq : q,
assume Hp : p,
have H1 : q → (r → s),
  from H Hp,
have H2 : r → s,
  from H1 Hq,
show s,
  from H2 Hr

-- 5ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
assume Hr : r,
assume Hq : q,
assume Hp : p,
have H1 : q → (r → s),
  from H Hp,
have H2 : r → s,
  from H1 Hq,
H2 Hr

-- 6ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
assume Hr : r,
assume Hq : q,
assume Hp : p,
have H1 : q → (r → s),
  from H Hp,

```



```

(H1 Hq) Hr

-- 7ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
assume Hr : r,
assume Hq : q,
assume Hp : p,
((H Hp) Hq) Hr

-- 8ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
λ Hr Hq Hp, ((H Hp) Hq) Hr

-- 9ª demostración
example
  (H : p → (q → (r → s)))
  : r → (q → (p → s)) :=
-- by hint
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.11. $\vdash (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$

```

-----
-- Ejercicio. Demostrar
--    $\vdash (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
begin
  intros Hpq Hp,
  apply Hpqr,

```

```

{ exact Hp, },
{ apply Hpq,
  exact Hp, },
end

-- 2ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
begin
  intros Hpq Hp,
  apply Hpqr,
  { exact Hp, },
  { exact Hpq Hp, },
end

-- 3ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
begin
  intros Hpq Hp,
  exact Hpqr Hp (Hpq Hp),
end

-- 4ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
λ Hpq Hp, Hpqr Hp (Hpq Hp)

-- 5ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,
have Hq : q,
  from Hpq Hp,
have Hqr : q → r,
  from Hpqr Hp,
show r,
  from Hqr Hq

-- 6ª demostración

```

```

example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,
have Hq : q,
  from Hpq Hp,
have Hqr : q → r,
  from Hpqr Hp,
Hqr Hq

-- 7ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,
have Hq : q,
  from Hpq Hp,
(Hpqr Hp) Hq

-- 8ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
assume Hpq : p → q,
assume Hp : p,
(Hpqr Hp) (Hpq Hp)

-- 9ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
λ Hpq Hp, (Hpqr Hp) (Hpq Hp)

-- 10ª demostración
example
  (Hpqr : p → (q → r))
  : (p → q) → (p → r) :=
-- by hint
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 2.12. $(p \rightarrow q) \rightarrow r \vdash p \rightarrow (q \rightarrow r)$

```
-- Ejercicio. Demostrar
-- (p → q) → r ⊢ p → (q → r)
```

```
import tactic
variables (p q r : Prop)
```

```
-- 1ª demostración
```

```
example
  (Hpqr : (p → q) → r)
  : p → (q → r) :=
```

```
begin
  intros Hp Hq,
  apply Hpqr,
  intro Hp,
  exact Hq,
```

```
end
```

```
-- 2ª demostración
```

```
example
  (Hpqr : (p → q) → r)
  : p → (q → r) :=
```

```
begin
  intros Hp Hq,
  apply Hpqr,
  exact (λ Hp, Hq),
```

```
end
```

```
-- 3ª demostración
```

```
example
  (Hpqr : (p → q) → r)
  : p → (q → r) :=
```

```
begin
  intros Hp Hq,
  exact Hpqr (λ Hp, Hq),
```

```
end
```

```
-- 4ª demostración
```

```
example
  (Hpqr : (p → q) → r)
  : p → (q → r) :=
```

```
λ Hp Hq, Hpqr (λ Hp, Hq)
```

```
-- 5ª demostración
```

```
example
```

```
(Hpqr : (p → q) → r)
: p → (q → r) :=
```

```
assume Hp : p,
```

```
assume Hq : q,
```

```
have Hpq : p → q,
```

```
{ assume p,
  show q,
  from Hq },
```

```
show r,
```

```
from Hpqr Hpq
```

```
-- 6ª demostración
```

```
example
```

```
(Hpqr : (p → q) → r)
: p → (q → r) :=
```

```
assume Hp : p,
```

```
assume Hq : q,
```

```
have Hpq : p → q,
```

```
{ assume p,
  show q,
  from Hq },
```

```
Hpqr Hpq
```

```
-- 7ª demostración
```

```
example
```

```
(Hpqr : (p → q) → r)
: p → (q → r) :=
```

```
assume Hp : p,
```

```
assume Hq : q,
```

```
have Hpq : p → q,
```

```
from (λ p, Hq),
```

```
Hpqr Hpq
```

```
-- 8ª demostración
```

```
example
```

```
(Hpqr : (p → q) → r)
: p → (q → r) :=
```

```
assume Hp : p,
```

```
assume Hq : q,
```

```
Hpqr (λ p, Hq)
```

```
-- 9ª demostración
example
  (Hpqr : (p → q) → r)
  : p → (q → r) :=
λ Hp Hq, Hpqr (λ p, Hq)

-- 10ª demostración
example
  (Hpqr : (p → q) → r)
  : p → (q → r) :=
-- by hint
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

# Capítulo 3

## Ejercicios sobre conjunciones

### 3.1. $p, q \vdash p \wedge q$

```
-- Ejercicio. Demostrar
--    $p, q \vdash p \wedge q$ 
```

```
import tactic
variables (p q : Prop)
```

```
-- 1ª demostración
```

```
example
  (Hp : p)
  (Hq : q)
  : p ∧ q :=
```

```
begin
  split,
  { exact Hp, },
  { exact Hq, },
end
```

```
-- 2ª demostración
```

```
example
  (Hp : p)
  (Hq : q)
  : p ∧ q :=
and.intro Hp Hq
```

```
-- 3ª demostración
```

```
example
```

```

(Hp : p)
(Hq : q)
: p ∧ q :=
-- by library_search
⟨Hp, Hq⟩

-- 4ª demostración
example
(Hp : p)
(Hq : q)
: p ∧ q :=
-- by hint
by tauto

-- 5ª demostración
example
(Hp : p)
(Hq : q)
: p ∧ q :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 3.2. $p \wedge q \vdash p$

```

-----
-- Ejercicio. Demostrar
--    $p \wedge q \vdash p$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
(H : p ∧ q)
: p :=
begin
  cases H with Hp Hq,
  exact Hp,
end

-- 2ª demostración

```



```
example
  (H : p ∧ q)
  : p :=
and.elim_left H

-- 3ª demostración
example
  (H : p ∧ q)
  : p :=
and.left H

-- 4ª demostración
example
  (H : p ∧ q)
  : p :=
H.left

-- 5ª demostración
example
  (H : p ∧ q)
  : p :=
H.1

-- 6ª demostración
example
  (H : p ∧ q)
  : p :=
-- by library_search
H.left

-- 7ª demostración
example
  (H : p ∧ q)
  : p :=
-- by hint
by tauto

-- 8ª demostración
example
  (H : p ∧ q)
  : p :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.3. $p \wedge q \vdash q$

```
-- Ejercicio. Demostrar  
--  $p \wedge q \vdash q$   
-----
```

```
import tactic  
variables (p q : Prop)
```

```
-- 1ª demostración
```

```
example  
  (H : p ∧ q)  
  : q :=  
begin  
  cases H with Hp Hq,  
  exact Hq,  
end
```

```
-- 2ª demostración
```

```
example  
  (H : p ∧ q)  
  : q :=  
and.elim_right H
```

```
-- 3ª demostración
```

```
example  
  (H : p ∧ q)  
  : q :=  
and.right H
```

```
-- 4ª demostración
```

```
example  
  (H : p ∧ q)  
  : q :=  
H.right
```

```
-- 5ª demostración
```

```
example  
  (H : p ∧ q)  
  : q :=  
H.2
```

```
-- 6ª demostración
```

```
example
  (H : p ∧ q)
  : q :=
-- by library_search
H.right
```

```
-- 7ª demostración
```

```
example
  (H : p ∧ q)
  : q :=
-- by hint
by tauto
```

```
-- 8ª demostración
```

```
example
  (H : p ∧ q)
  : q :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.4. $p \wedge (q \wedge r) \vdash (p \wedge q) \wedge r$

```
-----
-- Ejercicio. Demostrar
--    $p \wedge (q \wedge r) \vdash (p \wedge q) \wedge r$ 
-----
```

```
import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
begin
  cases Hpqr with Hp Hqr,
  cases Hqr with Hq Hr,
  split,
  { split,
    { exact Hp, },
    { exact Hq, }},
  { exact Hr, },
```

```

end

-- 2ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
begin
  cases Hpqr with Hp Hqr,
  cases Hqr with Hq Hr,
  split,
  { exact ⟨Hp, Hq⟩, },
  { exact Hr, },
end

-- 3ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
begin
  cases Hpqr with Hp Hqr,
  cases Hqr with Hq Hr,
  exact ⟨⟨Hp, Hq⟩, Hr⟩,
end

-- 4ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
begin
  rcases Hpqr with ⟨Hp, ⟨Hq, Hr⟩⟩,
  exact ⟨⟨Hp, Hq⟩, Hr⟩,
end

-- 5ª demostración
example :
  p ∧ (q ∧ r) → (p ∧ q) ∧ r :=
begin
  rintros ⟨Hp, ⟨Hq, Hr⟩⟩,
  exact ⟨⟨Hp, Hq⟩, Hr⟩,
end

-- 6ª demostración
example :
  p ∧ (q ∧ r) → (p ∧ q) ∧ r :=
λ ⟨Hp, ⟨Hq, Hr⟩⟩, ⟨⟨Hp, Hq⟩, Hr⟩

```

```

-- 7ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
have Hp : p,
  from and.left Hpqr,
have Hqr : q ∧ r,
  from and.right Hpqr,
have Hq : q,
  from and.left Hqr,
have Hr : r,
  from and.right Hqr,
have Hpq : p ∧ q,
  from and.intro Hp Hq,
show (p ∧ q) ∧ r,
  from and.intro Hpq Hr

-- 8ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
-- by library_search
(and_assoc p q).mpr Hpqr

-- 9ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
-- by hint
by tauto

-- 10ª demostración
example
  (Hpqr : p ∧ (q ∧ r))
  : (p ∧ q) ∧ r :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.5. $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$

```

-----
-- Ejercicio. Demostrar
--    $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (Hpqr : (p ∧ q) ∧ r)
  : p ∧ (q ∧ r) :=
begin
  rcases Hpqr with ⟨⟨Hp, Hq⟩, Hr⟩,
  exact ⟨Hp, ⟨Hq, Hr⟩⟩,
end

-- 2ª demostración
example
  : (p ∧ q) ∧ r → p ∧ (q ∧ r) :=
begin
  rintro ⟨⟨Hp, Hq⟩, Hr⟩,
  exact ⟨Hp, ⟨Hq, Hr⟩⟩,
end

-- 3ª demostración
example
  : (p ∧ q) ∧ r → p ∧ (q ∧ r) :=
λ ⟨⟨Hp, Hq⟩, Hr⟩, ⟨Hp, ⟨Hq, Hr⟩⟩

-- 4ª demostración
example
  (Hpqr : (p ∧ q) ∧ r)
  : p ∧ (q ∧ r) :=
have Hpq : p ∧ q,
  from and.left Hpqr,
have Hr : r,
  from and.right Hpqr,
have Hp : p,
  from and.left Hpq,
have Hq : q,
  from and.right Hpq,

```

```

have Hqr : q ∧ r,
  from and.intro Hq Hr,
show p ∧ (q ∧ r),
  from and.intro Hp Hqr

-- 5ª demostración
example
  (Hpqr : (p ∧ q) ∧ r)
  : p ∧ (q ∧ r) :=
-- by library_search
(and_assoc p q).mp Hpqr

-- 6ª demostración
example
  (Hpqr : (p ∧ q) ∧ r)
  : p ∧ (q ∧ r) :=
-- by hint
by tauto

-- 7ª demostración
example
  (Hpqr : (p ∧ q) ∧ r)
  : p ∧ (q ∧ r) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.6. $p \wedge q \vdash p \rightarrow q$

```

-----
-- Ejercicio. Demostrar
--    $p \wedge q \vdash p \rightarrow q$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (Hpq : p ∧ q)
  : p → q :=
begin
  intro p,

```

```

    exact Hpq.right,
end

-- 2ª demostración
example
  (Hpq : p ∧ q)
  : p → q :=
λ _, Hpq.2

-- 3ª demostración
example
  (Hpq : p ∧ q)
  : p → q :=
assume Hp : p,
show q,
from and.right Hpq

-- 4ª demostración
example
  (Hpq : p ∧ q)
  : p → q :=
-- by hint
by tauto

-- 5ª demostración
example
  (Hpq : p ∧ q)
  : p → q :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.7. $(p \rightarrow q) \wedge (p \rightarrow r) \vdash p \rightarrow q \wedge r$

```

-----
-- Ejercicio. Demostrar
--    $(p \rightarrow q) \wedge (p \rightarrow r) \vdash p \rightarrow q \wedge r$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración

```



```

example
  (H : (p → q) ∧ (p → r))
  : p → q ∧ r :=
begin
  cases H with Hpq Hpr,
  intro Hp,
  split,
  { apply Hpq,
    exact Hp, },
  { apply Hpr,
    exact Hp, },
end

-- 2ª demostración
example
  (H : (p → q) ∧ (p → r))
  : p → q ∧ r :=
begin
  cases H with Hpq Hpr,
  intro Hp,
  split,
  { exact Hpq Hp, },
  { exact Hpr Hp, },
end

-- 3ª demostración
example
  (H : (p → q) ∧ (p → r))
  : p → q ∧ r :=
begin
  cases H with Hpq Hpr,
  intro Hp,
  exact ⟨Hpq Hp, Hpr Hp⟩,
end

-- 4ª demostración
example
  : (p → q) ∧ (p → r) → (p → q ∧ r) :=
begin
  rintros ⟨Hpq, Hpr⟩ Hp,
  exact ⟨Hpq Hp, Hpr Hp⟩,
end

-- 5ª demostración
example

```

```

: (p → q) ∧ (p → r) → (p → q ∧ r) :=
λ ⟨Hpq, Hpr⟩ Hp, ⟨Hpq Hp, Hpr Hp⟩

```

```
-- 6ª demostración
```

```
example
```

```

  (H : (p → q) ∧ (p → r))
  : p → q ∧ r :=
have Hpq : p → q,
  from and.left H,
have Hpr : p → r,
  from and.right H,
assume Hp : p,
have Hq : q,
  from Hpq Hp,
have Hr : r,
  from Hpr Hp,
show q ∧ r,
  from and.intro Hq Hr

```

```
-- 7ª demostración
```

```
example
```

```

  (H : (p → q) ∧ (p → r))
  : p → q ∧ r :=
-- by library_search
imp_and_distrib.mpr H

```

```
-- 8ª demostración
```

```
example
```

```

  (H : (p → q) ∧ (p → r))
  : p → q ∧ r :=
-- by hint
by tauto

```

```
-- 9ª demostración
```

```
example
```

```

  (H : (p → q) ∧ (p → r))
  : p → q ∧ r :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

**3.8.  $p \rightarrow q \wedge r \vdash (p \rightarrow q) \wedge (p \rightarrow r)$** 

```

-----
-- Ejercicio. Demostrar
--    $p \rightarrow q \wedge r \vdash (p \rightarrow q) \wedge (p \rightarrow r)$ 
-----

```

```

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : p → q ∧ r)
  : (p → q) ∧ (p → r) :=
begin
  split,
  { intro Hp,
    have Hqr : q ∧ r,
      from H Hp,
    exact Hqr.left, },
  { intro Hp,
    have Hqr : q ∧ r,
      from H Hp,
    exact Hqr.right, },
end

-- 2ª demostración
example
  (H : p → q ∧ r)
  : (p → q) ∧ (p → r) :=
begin
  split,
  { intro Hp,
    exact (H Hp).left, },
  { intro Hp,
    exact (H Hp).right, },
end

-- 3ª demostración
example
  (H : p → q ∧ r)
  : (p → q) ∧ (p → r) :=
<λ Hp, (H Hp).left,
  λ Hp, (H Hp).right>

```

```
-- 4ª demostración
example
  (H : p → q ∧ r)
  : (p → q) ∧ (p → r) :=
have Hpq : p → q, from
  assume Hp : p,
  have Hqr : q ∧ r,
  from H Hp,
  show q,
  from and.left Hqr,
have Hpr : p → r, from
  assume Hp : p,
  have Hqr : q ∧ r,
  from H Hp,
  show r,
  from and.right Hqr,
show (p → q) ∧ (p → r),
  from and.intro Hpq Hpr
```

```
-- 5ª demostración
example
  (H : p → q ∧ r)
  : (p → q) ∧ (p → r) :=
and.intro
  ( assume Hp : p,
    have Hqr : q ∧ r,
    from H Hp,
    show q,
    from and.left Hqr)
  ( assume Hp : p,
    have Hqr : q ∧ r,
    from H Hp,
    show r,
    from and.right Hqr)
```

```
-- 6ª demostración
example
  (H : p → q ∧ r)
  : (p → q) ∧ (p → r) :=
-- by library_search
imp_and_distrib.mp H
```

```
-- 7ª demostración
example
```

```

(H : p → q ∧ r)
: (p → q) ∧ (p → r) :=
-- by hint
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.9. $p \rightarrow (q \rightarrow r) \vdash p \wedge q \rightarrow r$

```

-----
-- Ejercicio Demostrar
--    $p \rightarrow (q \rightarrow r) \vdash p \wedge q \rightarrow r$ 
-----

```

```

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
begin
  intro Hpq,
  apply H,
  { exact Hpq.left, },
  { exact Hpq.right, },
end

-- 2ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
begin
  intro Hpq,
  exact (H Hpq.left) Hpq.right,
end

-- 3ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
λ Hpq, (H Hpq.left) Hpq.right

```

```

-- 4ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
λ Hpq, H Hpq.1 Hpq.2

-- 5ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
assume Hpq : p ∧ q,
have Hp : p,
  from and.left Hpq,
have Hq : q,
  from and.right Hpq,
have Hqr : q → r,
  from H Hp,
show r,
  from Hqr Hq

-- 6ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
-- by library_search
and_imp.mpr H

-- 7ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
-- by hint
by tauto

-- 8ª demostración
example
  (H : p → (q → r))
  : p ∧ q → r :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

**3.10.  $p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$** 

```
-- Ejercicio. Demostrar  
--  $p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$ 
```

```
import tactic  
variables (p q r : Prop)
```

```
-- 1ª demostración
```

```
example  
  (H : p  $\wedge$  q  $\rightarrow$  r)  
  : p  $\rightarrow$  (q  $\rightarrow$  r) :=  
begin  
  intros Hp Hq,  
  apply H,  
  split,  
  { exact Hp, },  
  { exact Hq, },  
end
```

```
-- 2ª demostración
```

```
example  
  (H : p  $\wedge$  q  $\rightarrow$  r)  
  : p  $\rightarrow$  (q  $\rightarrow$  r) :=  
begin  
  intros Hp Hq,  
  apply H,  
  exact ⟨Hp, Hq⟩,  
end
```

```
-- 3ª demostración
```

```
example  
  (H : p  $\wedge$  q  $\rightarrow$  r)  
  : p  $\rightarrow$  (q  $\rightarrow$  r) :=  
begin  
  intros Hp Hq,  
  exact H ⟨Hp, Hq⟩,  
end
```

```
-- 4ª demostración
```

```
example  
  (H : p  $\wedge$  q  $\rightarrow$  r)
```

```

    : p → (q → r) :=
  λ Hp Hq, H ⟨Hp, Hq⟩

-- 5ª demostración
example
  (H : p ∧ q → r)
  : p → (q → r) :=
assume Hp : p,
show q → r, from
  assume Hq : q,
  have Hpq : p ∧ q,
    from and.intro Hp Hq,
  show r,
    from H Hpq

-- 6ª demostración
example
  (H : p ∧ q → r)
  : p → (q → r) :=
-- by library_search
and_imp.mp H

-- 7ª demostración
example
  (H : p ∧ q → r)
  : p → (q → r) :=
-- by hint
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.11. $(p \rightarrow q) \rightarrow r \vdash p \wedge q \rightarrow r$

```

-----
-- Ejercicio. Demostrar
--    $(p \rightarrow q) \rightarrow r \vdash p \wedge q \rightarrow r$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example

```



```

(H : (p → q) → r)
: p ∧ q → r :=
begin
  intro Hpq,
  apply H,
  intro Hp,
  exact Hpq.right,
end

-- 2ª demostración
example
(H : (p → q) → r)
: p ∧ q → r :=
begin
  intro Hpq,
  apply H,
  exact (λ Hp, Hpq.right),
end

-- 3ª demostración
example
(H : (p → q) → r)
: p ∧ q → r :=
begin
  intro Hpq,
  exact H (λ Hp, Hpq.right),
end

-- 4ª demostración
example
(H : (p → q) → r)
: p ∧ q → r :=
λ Hpq, H (λ _, Hpq.right)

-- 5ª demostración
example
(H : (p → q) → r)
: p ∧ q → r :=
assume Hpq : p ∧ q,
have H1 : p → q, from
  assume Hp : p,
  show q,
  from and.right Hpq,
show r,
from H H1

```

```
-- 6ª demostración
example
  (H : (p → q) → r)
  : p ∧ q → r :=
-- by hint
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 3.12. $p \wedge (q \rightarrow r) \vdash (p \rightarrow q) \rightarrow r$

```
-- Ejercicio. Demostrar
--    $p \wedge (q \rightarrow r) \vdash (p \rightarrow q) \rightarrow r$ 
```

```
import tactic
variables (p q r : Prop)
```

```
-- 1ª demostración
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
begin
  intro Hpq,
  cases H with Hp Hqr,
  apply Hqr,
  apply Hpq,
  exact Hp,
end
```

```
-- 2ª demostración
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
begin
  intro Hpq,
  cases H with Hp Hqr,
  apply Hqr,
  exact Hpq Hp,
end
```

```
-- 3ª demostración
```

```
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
begin
  intro Hpq,
  cases H with Hp Hqr,
  exact Hqr (Hpq Hp),
end
```

-- 4ª demostración

```
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
begin
  intro Hpq,
  exact H.2 (Hpq H.1),
end
```

-- 5ª demostración

```
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
λ Hpq, H.2 (Hpq H.1)
```

-- 6ª demostración

```
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
assume Hpq : p → q,
have Hp : p,
  from and.left H,
have Hq : q,
  from Hpq Hp,
have Hqr : q → r,
  from H.right,
show r,
  from Hqr Hq
```

-- 7ª demostración

```
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
-- by hint
by tauto
```

```
-- 8ª demostració
example
  (H : p ∧ (q → r))
  : (p → q) → r :=
-- by hint
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

# Capítulo 4

## Ejercicios sobre disyunciones

### 4.1. $p \vdash p \vee q$

```
-- Ejercicio. Demostrar
--    $p \vdash p \vee q$ 
```

```
import tactic
variables (p q : Prop)
```

```
-- 1ª demostración
```

```
example
  (H : p)
  : p  $\vee$  q :=
begin
  left,
  exact H,
end
```

```
-- 2ª demostración
```

```
example
  (H : p)
  : p  $\vee$  q :=
or.intro_left q H
```

```
-- 3ª demostración
```

```
example
  (H : p)
  : p  $\vee$  q :=
-- by library_search
```

```

or.inl H

-- 4ª demostración
example
  (H : p)
  : p ∨ q :=
-- by hint
by tauto

-- 5ª demostración
example
  (H : p)
  : p ∨ q :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.2. $q \vdash p \vee q$

```

-----
-- Ejercicio. Demostrar
--    $q \vdash p \vee q$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (H : q)
  : p ∨ q :=
begin
  right,
  exact H,
end

-- 2ª demostración
example
  (H : q)
  : p ∨ q :=
or.intro_right p H

-- 3ª demostración

```

```

example
  (H : q)
  : p ∨ q :=
-- by library_search
or.inr H

-- 4ª demostración
example
  (H : q)
  : p ∨ q :=
-- by hint
by tauto

-- 5ª demostración
example
  (H : q)
  : p ∨ q :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.3. $p \vee q \vdash q \vee p$

```

-----
-- Ejercicio. Demostrar
--    $p \vee q \vdash q \vee p$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
begin
  cases H with Hp Hq,
  { right,
    exact Hp, },
  { left,
    exact Hq, },
end

```

```
-- 2ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
begin
  cases H with Hp Hq,
  { exact or.inr Hp, },
  { exact or.inl Hq, },
end
```

```
-- 3ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
or.elim H
( assume Hp : p,
  show q ∨ p,
  from or.inr Hp)
( assume Hq : q,
  show q ∨ p,
  from or.inl Hq)
```

```
-- 4ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
or.elim H
( assume Hp : p,
  or.inr Hp)
( assume Hq : q,
  or.inl Hq)
```

```
-- 5ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
or.elim H
( λ Hp, or.inr Hp)
( λ Hq, or.inl Hq)
```

```
-- 6ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
or.elim H or.inr or.inl
```



```

-- 7ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
-- by library_search
or.swap H

-- 8ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
-- by hint
by tauto

-- 9ª demostración
example
  (H : p ∨ q)
  : q ∨ p :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.4. $q \rightarrow r \vdash p \vee q \rightarrow p \vee r$

```

-----
-- Ejercicio. Demostrar
--    $q \rightarrow r \vdash p \vee q \rightarrow p \vee r$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
begin
  intro H1,
  cases H1 with Hp Hq,
  { left,
    exact Hp, },
  { right,
    apply H,

```

```

    exact Hq, },
end

-- 2ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
begin
  rintro (Hp | Hq),
  { left,
    exact Hp, },
  { right,
    exact H Hq, },
end

-- 3ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
begin
  rintro (Hp | Hq),
  { exact or.inl Hp, },
  { exact or.inr (H Hq), },
end

-- 4ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
assume H1 : p ∨ q,
or.elim H1
( assume Hp : p,
  show p ∨ r,
  from or.inl Hp)
( assume Hq : q,
  have Hr : r,
  from H Hq,
  show p ∨ r,
  from or.inr Hr)

-- 5ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
assume H1 : p ∨ q,

```

```

or.elim H1
  ( assume Hp : p,
    or.inl Hp)
  ( assume Hq : q,
    have Hr : r,
      from H Hq,
    or.inr Hr)

-- 6ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
assume H1 : p ∨ q,
or.elim H1
  ( assume Hp : p,
    or.inl Hp)
  ( assume Hq : q,
    or.inr (H Hq))

-- 7ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
assume H1 : p ∨ q,
or.elim H1
  ( λ Hp, or.inl Hp)
  ( λ Hq, or.inr (H Hq))

-- 8ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
assume H1 : p ∨ q,
or.elim H1
  or.inl
  ( λ Hq, or.inr (H Hq))

-- 9ª demostración
example
  (H : q → r)
  : p ∨ q → p ∨ r :=
λ H1, or.elim H1 or.inl (λ Hq, or.inr (H Hq))

-- 10ª demostración
example

```

```

(H : q → r)
: p ∨ q → p ∨ r :=
-- by library_search
or.imp_right H

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.5. $p \vee p \vdash p$

```

-- Ejercicio. Demostrar
--   p ∨ p ⊢ p
-- -----

```

```

import tactic
variable (p : Prop)

-- 1ª demostración
example
  (H : p ∨ p)
  : p :=
begin
  cases H with Hp Hp,
  { exact Hp, },
  { exact Hp, },
end

-- 2ª demostración
example
  (H : p ∨ p)
  : p :=
by cases H ; assumption

-- 3ª demostración
example
  (H : p ∨ p)
  : p :=
or.elim H
  ( assume Hp : p,
    show p,
    from Hp)
  ( assume Hp : p,
    show p,

```

```
      from Hp)

-- 4ª demostración
example
  (H : p ∨ p)
  : p :=
or.elim H
  ( assume Hp : p,
    Hp)
  ( assume Hp : p,
    Hp)

-- 5ª demostración
example
  (H : p ∨ p)
  : p :=
or.elim H
  ( λ Hp, Hp)
  ( λ Hp, Hp)

-- 6ª demostración
example
  (H : p ∨ p)
  : p :=
or.elim H id id

-- 7ª demostración
example
  (H : p ∨ p)
  : p :=
-- by library_search
(or_self p).mp H

-- 8ª demostración
example
  (H : p ∨ p)
  : p :=
-- by hint
by tauto

-- 9ª demostración
example
  (H : p ∨ p)
  : p :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.6. $p \vdash p \vee p$

```
-- Ejercicio. Demostrar
--    $p \vdash p \vee p$ 
```

```
import tactic
variables (p : Prop)
```

```
-- 1ª demostración
example
  (H : p)
  : p  $\vee$  p :=
-- by library_search
or.inl H
```

```
-- 2ª demostración
example
  (H : p)
  : p  $\vee$  p :=
-- by hint
by tauto
```

```
-- 3ª demostración
example
  (H : p)
  : p  $\vee$  p :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.7. $p \vee (q \vee r) \vdash (p \vee q) \vee r$

```
-- Ejercicio. Demostrar
--    $p \vee (q \vee r) \vdash (p \vee q) \vee r$ 
```

```

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
begin
  cases H with Hp Hqr,
  { left,
    left,
    exact Hp, },
  { cases Hqr with Hq Hr,
    { left,
      right,
      exact Hq, },
    { right,
      exact Hr, }},
end

-- 2ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
or.elim H
  ( assume Hp : p,
    have Hpq : p ∨ q,
      from or.inl Hp,
    show (p ∨ q) ∨ r,
      from or.inl Hpq)
  ( assume Hqr : q ∨ r,
    show (p ∨ q) ∨ r, from
      or.elim Hqr
        ( assume Hq : q,
          have Hpq : p ∨ q,
            from or.inr Hq,
          show (p ∨ q) ∨ r,
            from or.inl Hpq)
        ( assume Hr : r,
          show (p ∨ q) ∨ r,
            from or.inr Hr))

-- 3ª demostración
example
  (H : p ∨ (q ∨ r))

```

```

: (p ∨ q) ∨ r :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,
  from or.inl Hp,
  show (p ∨ q) ∨ r,
  from or.inl Hpq)
( assume Hqr : q ∨ r,
  show (p ∨ q) ∨ r, from
  or.elim Hqr
    ( assume Hq : q,
      have Hpq : p ∨ q,
      from or.inr Hq,
      or.inl Hpq)
    ( assume Hr : r,
      or.inr Hr))

```

-- 4ª demostración

example

```

(H : p ∨ (q ∨ r))
: (p ∨ q) ∨ r :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,
  from or.inl Hp,
  show (p ∨ q) ∨ r,
  from or.inl Hpq)
( assume Hqr : q ∨ r,
  show (p ∨ q) ∨ r, from
  or.elim Hqr
    ( assume Hq : q,
      have Hpq : p ∨ q,
      from or.inr Hq,
      or.inl Hpq)
    or.inr)

```

-- 5ª demostración

example

```

(H : p ∨ (q ∨ r))
: (p ∨ q) ∨ r :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,
  from or.inl Hp,
  show (p ∨ q) ∨ r,

```



```

    from or.inl Hpq)
  ( assume Hqr : q ∨ r,
    show (p ∨ q) ∨ r, from
      or.elim Hqr
        ( λ Hq, or.inl (or.inr Hq))
      or.inr)

-- 6ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
or.elim H
  ( assume Hp : p,
    have Hpq : p ∨ q,
      from or.inl Hp,
    show (p ∨ q) ∨ r,
      from or.inl Hpq)
  ( λ Hqr, or.elim Hqr ( λ Hq, or.inl (or.inr Hq)) or.inr)

-- 7ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
or.elim H
  ( assume Hp : p,
    have Hpq : p ∨ q,
      from or.inl Hp,
    or.inl Hpq)
  (λ Hqr, or.elim Hqr ( λ Hq, or.inl (or.inr Hq)) or.inr)

-- 8ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
or.elim H
  ( assume Hp : p,
    or.inl (or.inl Hp))
  (λ Hqr, or.elim Hqr ( λ Hq, or.inl (or.inr Hq)) or.inr)

-- 9ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
or.elim H
  (λ Hp, or.inl (or.inl Hp))

```

```

(λ Hqr, or.elim Hqr ( λ Hq, or.inl (or.inr Hq)) or.inr)

-- 10ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
-- by library_search
or.assoc.mpr H

-- 11ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
-- by hint
by tauto

-- 12ª demostración
example
  (H : p ∨ (q ∨ r))
  : (p ∨ q) ∨ r :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.8. $(p \vee q) \vee r \vdash p \vee (q \vee r)$

```

-----
-- Ejercicio. Demostrar
--    $(p \vee q) \vee r \vdash p \vee (q \vee r)$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : (p ∨ q) ∨ r)
  : p ∨ (q ∨ r) :=
begin
  rcases H with ((Hp | Hq) | Hr),
  { left,
    exact Hp, },
  { right,

```

```

    left,
    exact Hq, },
  { right,
    right,
    exact Hr, },
end

-- 2ª demostración
example
  (H : (p ∨ q) ∨ r)
  : p ∨ (q ∨ r) :=
or.elim H
  ( assume Hpq : p ∨ q,
    show p ∨ q ∨ r, from
      or.elim Hpq
        ( assume Hp : p,
          show p ∨ (q ∨ r),
            from or.inl Hp)
        ( assume Hq : q,
          have Hqr: q ∨ r,
            from or.inl Hq,
          show p ∨ (q ∨ r),
            from or.inr Hqr))
  ( assume Hr : r,
    have Hqr: q ∨ r,
      from or.inr Hr,
    show p ∨ (q ∨ r),
      from or.inr Hqr)

-- 3ª demostración
example
  (H : (p ∨ q) ∨ r)
  : p ∨ (q ∨ r) :=
or.elim H
  ( λ Hpq, or.elim Hpq or.inl (λ Hq, or.inr (or.inl Hq)))
  ( λ Hr, or.inr (or.inr Hr))

-- 4ª demostración
example
  (H : (p ∨ q) ∨ r)
  : p ∨ (q ∨ r) :=
-- by library_search
or.assoc.mp H

-- 5ª demostración

```

```
example
  (H : (p ∨ q) ∨ r)
  : p ∨ (q ∨ r) :=
-- by hint
by tauto
```

```
-- 6ª demostración
```

```
example
  (H : (p ∨ q) ∨ r)
  : p ∨ (q ∨ r) :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.9. $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$

```
-- -----
-- Ejercicio. Demostrar
--    $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$ 
-- -----
```

```
import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : p ∧ (q ∨ r))
  : (p ∧ q) ∨ (p ∧ r) :=
begin
  cases H with Hp Hqr,
  cases Hqr with Hq Hr,
  { left,
    split,
    { exact Hp, },
    { exact Hq, }},
  { right,
    split,
    { exact Hp, },
    { exact Hr, }},
end

-- 2ª demostración
example
```

```

(H : p ∧ (q ∨ r))
: (p ∧ q) ∨ (p ∧ r) :=
begin
  cases H with Hp Hqr,
  cases Hqr with Hq Hr,
  { left,
    exact ⟨Hp, Hq⟩, },
  { right,
    exact ⟨Hp, Hr⟩, },
end

-- 3ª demostración
example
  (H : p ∧ (q ∨ r))
  : (p ∧ q) ∨ (p ∧ r) :=
  have Hp : p,
    from and.left H,
  have Hqr : q ∨ r,
    from and.right H,
  or.elim Hqr
  ( assume Hq : q,
    have Hpq : p ∧ q,
      from and.intro Hp Hq,
    show (p ∧ q) ∨ (p ∧ r),
      from or.inl Hpq)
  ( assume Hr : r,
    have Hpr : p ∧ r,
      from and.intro Hp Hr,
    show (p ∧ q) ∨ (p ∧ r),
      from or.inr Hpr)

-- 4ª demostración
example
  (H : p ∧ (q ∨ r))
  : (p ∧ q) ∨ (p ∧ r) :=
  or.elim H.2
  (λ Hq, or.inl ⟨H.1, Hq⟩)
  (λ Hr, or.inr ⟨H.1, Hr⟩)

-- 5ª demostración
example
  (H : p ∧ (q ∨ r))
  : (p ∧ q) ∨ (p ∧ r) :=
  -- by library_search
  and_or_distrib_left.mp H

```

```
-- 6ª demostración
example
  (H : p ∧ (q ∨ r))
  : (p ∧ q) ∨ (p ∧ r) :=
-- by hint
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.10. $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$

```
-----
-- Ejercicio. Demostrar
--   (p ∧ q) ∨ (p ∧ r) ⊢ p ∧ (q ∨ r)
-----
```

```
import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : (p ∧ q) ∨ (p ∧ r))
  : p ∧ (q ∨ r) :=
begin
  rcases H with (⟨Hp, Hq⟩ | ⟨Hp, Hr⟩),
  { exact ⟨Hp, or.inl Hq⟩, },
  { exact ⟨Hp, or.inr Hr⟩, },
end
```

```
-- 2ª demostración
example
  (H : (p ∧ q) ∨ (p ∧ r))
  : p ∧ (q ∨ r) :=
or.elim H
( assume Hpq : p ∧ q,
  have Hp : p,
    from and.left Hpq,
  have Hq : q,
    from and.right Hpq,
  have Hqr : q ∨ r,
    from or.inl Hq,
  show p ∧ (q ∨ r),
    from and.intro Hp Hqr)
```

```

( assume Hpr : p ∧ r,
  have Hp : p,
    from and.left Hpr,
  have Hr : r,
    from and.right Hpr,
  have Hqr : q ∨ r,
    from or.inr Hr,
  show p ∧ (q ∨ r),
    from and.intro Hp Hqr)

-- 3ª demostración
example
  (H : (p ∧ q) ∨ (p ∧ r))
  : p ∧ (q ∨ r) :=
or.elim H
  ( assume ⟨Hp, Hq⟩,
    have Hqr : q ∨ r,
      from or.inl Hq,
    show p ∧ (q ∨ r),
      from and.intro Hp Hqr)
  ( assume ⟨Hp, Hr⟩,
    have Hqr : q ∨ r,
      from or.inr Hr,
    show p ∧ (q ∨ r),
      from and.intro Hp Hqr)

-- 4ª demostración
example
  (H : (p ∧ q) ∨ (p ∧ r))
  : p ∧ (q ∨ r) :=
or.elim H
  (λ ⟨Hp, Hq⟩, ⟨Hp, or.inl Hq⟩)
  (λ ⟨Hp, Hr⟩, ⟨Hp, or.inr Hr⟩)

-- 5ª demostración
example
  (H : (p ∧ q) ∨ (p ∧ r))
  : p ∧ (q ∨ r) :=
-- by library_search
and_or_distrib_left.mpr H

-- 6ª demostración
example
  (H : (p ∧ q) ∨ (p ∧ r))
  : p ∧ (q ∨ r) :=

```

```
-- by hint
by tauto

-- 7ª demostración
example
  (H : (p ∧ q) ∨ (p ∧ r))
  : p ∧ (q ∨ r) :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.11. $p \vee (q \wedge r) \vdash (p \vee q) \wedge (p \vee r)$

```
-- Ejercicio. Demostrar
--    $p \vee (q \wedge r) \vdash (p \vee q) \wedge (p \vee r)$ 
```

```
import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
begin
  cases H with Hp Hqr,
  { split,
    { left,
      exact Hp, },
    { left,
      exact Hp, }},
  { split,
    { right,
      exact Hqr.left, },
    { right,
      exact Hqr.right, }},
end

-- 2ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
```



```

begin
  cases H with Hp Hqr,
  { split,
    { exact or.inl Hp, },
    { exact or.inl Hp, }},
  { split,
    { exact or.inr Hqr.left, },
    { exact or.inr Hqr.right, }},
end

-- 3ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
begin
  cases H with Hp Hqr,
  { exact ⟨or.inl Hp, or.inl Hp⟩, },
  { exact ⟨or.inr Hqr.left, or.inr Hqr.right⟩, },
end

-- 4ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
  (λ Hp, ⟨or.inl Hp, or.inl Hp⟩)
  (λ Hqr, ⟨or.inr Hqr.1, or.inr Hqr.2⟩)

-- 5ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
  (λ h, ⟨or.inl h, or.inl h⟩)
  (λ h, ⟨or.inr h.1, or.inr h.2⟩)

-- 6ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
  (
    assume Hp : p,
    have Hpq : p ∨ q,
    from or.inl Hp,
    have Hpr : p ∨ r,

```

```

    from or.inl Hp,
  show (p ∨ q) ∧ (p ∨ r),
    from and.intro Hpq Hpr)
( assume Hqr : q ∧ r,
  have Hq : q,
    from and.left Hqr,
  have Hr : r,
    from and.right Hqr,
  have Hpq : p ∨ q,
    from or.inr Hq,
  have Hpr : p ∨ r,
    from or.inr Hr,
  show (p ∨ q) ∧ (p ∨ r),
    from and.intro Hpq Hpr)

```

-- 7ª demostración

example

```

  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,
    from or.inl Hp,
  have Hpr : p ∨ r,
    from or.inl Hp,
  show (p ∨ q) ∧ (p ∨ r),
    from and.intro Hpq Hpr)
( assume Hqr : q ∧ r,
  have Hq : q,
    from and.left Hqr,
  have Hr : r,
    from and.right Hqr,
  have Hpq : p ∨ q,
    from or.inr Hq,
  have Hpr : p ∨ r,
    from or.inr Hr,
  and.intro Hpq Hpr)

```

-- 8ª demostración

example

```

  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,

```

```

    from or.inl Hp,
    have Hpr : p  $\vee$  r,
    from or.inl Hp,
    show (p  $\vee$  q)  $\wedge$  (p  $\vee$  r),
    from and.intro Hpq Hpr)
( assume Hqr : q  $\wedge$  r,
  have Hq : q,
    from and.left Hqr,
  have Hr : r,
    from and.right Hqr,
  and.intro (or.inr Hq) (or.inr Hr))

-- 9ª demostración
example
  (H : p  $\vee$  (q  $\wedge$  r))
  : (p  $\vee$  q)  $\wedge$  (p  $\vee$  r) :=
or.elim H
( assume Hp : p,
  have Hpq : p  $\vee$  q,
    from or.inl Hp,
  have Hpr : p  $\vee$  r,
    from or.inl Hp,
  show (p  $\vee$  q)  $\wedge$  (p  $\vee$  r),
    from and.intro Hpq Hpr)
( assume Hqr : q  $\wedge$  r,
  and.intro (or.inr (and.left Hqr)) (or.inr (and.right Hqr)))

-- 10ª demostración
example
  (H : p  $\vee$  (q  $\wedge$  r))
  : (p  $\vee$  q)  $\wedge$  (p  $\vee$  r) :=
or.elim H
( assume Hp : p,
  have Hpq : p  $\vee$  q,
    from or.inl Hp,
  have Hpr : p  $\vee$  r,
    from or.inl Hp,
  show (p  $\vee$  q)  $\wedge$  (p  $\vee$  r),
    from and.intro Hpq Hpr)
( assume Hqr : q  $\wedge$  r,
  and.intro (or.inr Hqr.1) (or.inr Hqr.2))

-- 11ª demostración
example
  (H : p  $\vee$  (q  $\wedge$  r))

```

```

: (p ∨ q) ∧ (p ∨ r) :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,
    from or.inl Hp,
  have Hpr : p ∨ r,
    from or.inl Hp,
  show (p ∨ q) ∧ (p ∨ r),
    from and.intro Hpq Hpr)
( assume Hqr : q ∧ r,
  ⟨or.inr Hqr.1, or.inr Hqr.2⟩)

-- 12ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,
    from or.inl Hp,
  have Hpr : p ∨ r,
    from or.inl Hp,
  show (p ∨ q) ∧ (p ∨ r),
    from and.intro Hpq Hpr)
( λ Hqr, ⟨or.inr Hqr.1, or.inr Hqr.2⟩)

-- 13ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
( assume Hp : p,
  have Hpq : p ∨ q,
    from or.inl Hp,
  have Hpr : p ∨ r,
    from or.inl Hp,
  and.intro Hpq Hpr)
( λ Hqr, ⟨or.inr Hqr.1, or.inr Hqr.2⟩)

-- 14ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
( assume Hp : p,

```

```

    have Hpq : p ∨ q,
      from or.inl Hp,
    have Hpr : p ∨ r,
      from or.inl Hp,
    ⟨Hpq, Hpr⟩)
  ( λ Hqr, ⟨or.inr Hqr.1, or.inr Hqr.2⟩)

-- 15ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
  ( assume Hp : p,
    ⟨or.inl Hp, or.inl Hp⟩)
  ( λ Hqr, ⟨or.inr Hqr.1, or.inr Hqr.2⟩)

-- 16ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
or.elim H
  ( λ Hp, ⟨or.inl Hp, or.inl Hp⟩)
  ( λ Hqr, ⟨or.inr Hqr.1, or.inr Hqr.2⟩)

-- 17ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
-- by library_search
or_and_distrib_left.mp H

-- 18ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
-- by hint
by tauto

-- 19ª demostración
example
  (H : p ∨ (q ∧ r))
  : (p ∨ q) ∧ (p ∨ r) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.12. $(p \vee q) \wedge (p \vee r) \vdash p \vee (q \wedge r)$

```

-----
-- Ejercicio. Demostrar
--    $(p \vee q) \wedge (p \vee r) \vdash p \vee (q \wedge r)$ 
-----

```

```

import tactic
variables (p q r : Prop)

```

```

-- 1ª demostración

```

```

example
  (H : (p ∨ q) ∧ (p ∨ r))
  : p ∨ (q ∧ r) :=

```

```

begin
  cases H with Hpq Hpr,
  cases Hpq with Hp Hq,
  { left,
    exact Hp, },
  { cases Hpr with Hp Hr,
    { left,
      exact Hp, },
    { right,
      split,
      { exact Hq, },
      { exact Hr, }}}},

```

```

end

```

```

-- 2ª demostración

```

```

example
  (H : (p ∨ q) ∧ (p ∨ r))
  : p ∨ (q ∧ r) :=

```

```

begin
  cases H with Hpq Hpr,
  cases Hpq with Hp Hq,
  { left,
    exact Hp, },
  { cases Hpr with Hp Hr,
    { left,
      exact Hp, },
    { right,
      exact ⟨Hq, Hr⟩, }},

```

```

end

```

```

-- 3ª demostración
example
  (H : (p ∨ q) ∧ (p ∨ r))
  : p ∨ (q ∧ r) :=
begin
  cases H with Hpq Hpr,
  cases Hpq with Hp Hq,
  { left,
    exact Hp, },
  { cases Hpr with Hp Hr,
    { left,
      exact Hp, },
    { exact or.inr ⟨Hq, Hr⟩, }},
end

-- 4ª demostración
example
  (H : (p ∨ q) ∧ (p ∨ r))
  : p ∨ (q ∧ r) :=
begin
  cases H with Hpq Hpr,
  cases Hpq with Hp Hq,
  { left,
    exact Hp, },
  { cases Hpr with Hp Hr,
    { exact or.inl Hp, },
    { exact or.inr ⟨Hq, Hr⟩, }},
end

-- 5ª demostración
example
  (H : (p ∨ q) ∧ (p ∨ r))
  : p ∨ (q ∧ r) :=
begin
  cases H with Hpq Hpr,
  cases Hpq with Hp Hq,
  { exact or.inl Hp, },
  { cases Hpr with Hp Hr,
    { exact or.inl Hp, },
    { exact or.inr ⟨Hq, Hr⟩, }},
end

-- 6ª demostración
example
  (H : (p ∨ q) ∧ (p ∨ r))

```

```

: p ∨ (q ∧ r) :=
begin
  rcases H with ⟨Hp | Hq, Hp | Hr⟩,
  { exact or.inl Hp, },
  { exact or.inl Hp, },
  { exact or.inl Hp, },
  { exact or.inr ⟨Hq, Hr⟩, },
end

```

-- 7ª demostración

example

```

(H : (p ∨ q) ∧ (p ∨ r))
: p ∨ (q ∧ r) :=
-- by library_search
or_and_distrib_left.mpr H

```

-- 8ª demostración

example

```

(H : (p ∨ q) ∧ (p ∨ r))
: p ∨ (q ∧ r) :=
have Hpq : p ∨ q,
  from and.left H,
or.elim Hpq
( assume Hp : p,
  show p ∨ (q ∧ r),
  from or.inl Hp )
( assume Hq : q,
  have Hpr : p ∨ r,
  from and.right H,
  or.elim Hpr
    ( assume Hp : p,
      show p ∨ (q ∧ r),
      from or.inl Hp )
    ( assume Hr : r,
      have Hqr : q ∧ r,
      from and.intro Hq Hr,
      show p ∨ (q ∧ r),
      from or.inr Hqr ))

```

-- 9ª demostración

example

```

(H : (p ∨ q) ∧ (p ∨ r))
: p ∨ (q ∧ r) :=
or.elim (and.left H)
or.inl

```



```

    (λ Hq, or.elim (and.right H)
      or.inl
      (λ Hr, or.inr ⟨Hq, Hr⟩))

-- 10ª demostración
example
  (H : (p ∨ q) ∧ (p ∨ r))
  : p ∨ (q ∧ r) :=
-- by hint
by tauto

-- 11ª demostración
example
  (H : (p ∨ q) ∧ (p ∨ r))
  : p ∨ (q ∧ r) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.13. $(p \rightarrow r) \wedge (q \rightarrow r) \vdash p \vee q \rightarrow r$

```

-----
-- Ejercicio. Demostrar
--    $(p \rightarrow r) \wedge (q \rightarrow r) \vdash p \vee q \rightarrow r$ 
-----

import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
begin
  cases H with Hpr Hqr,
  intro Hpq,
  cases Hpq with Hp Hq,
  { apply Hpr,
    exact Hp, },
  { apply Hqr,
    exact Hq, },
end

```

```
-- 2ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
begin
  cases H with Hpr Hqr,
  intro Hpq,
  cases Hpq with Hp Hq,
  { exact Hpr Hp, },
  { exact Hqr Hq, },
end
```

```
-- 3ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
begin
  intro Hpq,
  cases Hpq with Hp Hq,
  { exact H.left Hp, },
  { exact H.right Hq, },
end
```

```
-- 4ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
-- by library_search
or_imp_distrib.mpr H
```

```
-- 5ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
assume Hpq : p ∨ q,
or.elim Hpq
( assume Hp : p,
  have Hpr: p → r,
    from and.left H,
  show r,
    from Hpr Hp )
( assume Hq : q,
  have Hqr : q → r,
    from and.right H,
  show r,
```

```

    from Hqr Hq)

-- 6ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
assume Hpq : p ∨ q,
or.elim Hpq
  ( assume Hp : p,
    have Hpr: p → r,
      from and.left H,
    Hpr Hp )
  ( assume Hq : q,
    have Hqr : q → r,
      from and.right H,
    Hqr Hq)

-- 7ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
assume Hpq : p ∨ q,
or.elim Hpq
  ( assume Hp : p,
    H.1 Hp )
  ( assume Hq : q,
    H.2 Hq)

-- 8ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
assume Hpq : p ∨ q,
or.elim Hpq
  (λ Hp, H.1 Hp)
  (λ Hq, H.2 Hq)

-- 9ª demostración
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
assume Hpq : p ∨ q,
or.elim Hpq H.1 H.2

-- 10ª demostración

```

```
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
λ Hpq, or.elim Hpq H.1 H.2
```

```
-- 11ª demostración
```

```
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
-- by hint
by tauto
```

```
-- 12ª demostración
```

```
example
  (H : (p → r) ∧ (q → r))
  : p ∨ q → r :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 4.14. $p \vee q \rightarrow r \vdash (p \rightarrow r) \wedge (q \rightarrow r)$

```
-- -----
-- Ejercicio. Demostrar
--    $p \vee q \rightarrow r \vdash (p \rightarrow r) \wedge (q \rightarrow r)$ 
-- -----
```

```
import tactic
variables (p q r : Prop)

-- 1ª demostración
example
  (H : p ∨ q → r)
  : (p → r) ∧ (q → r) :=
begin
  split,
  { intro Hp,
    apply H,
    left,
    exact Hp, },
  { intro Hq,
    apply H,
    right,
```

```

    exact Hq, },
end

-- 2ª demostración
example
  (H : p ∨ q → r)
  : (p → r) ∧ (q → r) :=
begin
  split,
  { intro Hp,
    apply H,
    exact or.inl Hp, },
  { intro Hq,
    apply H,
    exact or.inr Hq, },
end

-- 3ª demostración
example
  (H : p ∨ q → r)
  : (p → r) ∧ (q → r) :=
begin
  split,
  { intro Hp,
    exact H (or.inl Hp), },
  { intro Hq,
    exact H (or.inr Hq), },
end

-- 4ª demostración
example
  (H : p ∨ q → r)
  : (p → r) ∧ (q → r) :=
<λ Hp, H (or.inl Hp),
  λ Hq, H (or.inr Hq)>

-- 5ª demostración
example
  (H : p ∨ q → r)
  : (p → r) ∧ (q → r) :=
-- by library_search
or_imp_distrib.mp H

-- 6ª demostración
example

```

```

(H : p ∨ q → r)
: (p → r) ∧ (q → r) :=
and.intro
  ( assume Hp : p,
    have Hpq : p ∨ q,
      from or.inl Hp,
    show r,
      from H Hpq)
  ( assume Hq : q,
    have Hpq : p ∨ q,
      from or.inr Hq,
    show r,
      from H Hpq)

```

-- 7ª demostración

example

```

(H : p ∨ q → r)
: (p → r) ∧ (q → r) :=
and.intro
  ( assume Hp : p,
    have Hpq : p ∨ q,
      from or.inl Hp,
    H Hpq)
  ( assume Hq : q,
    have Hpq : p ∨ q,
      from or.inr Hq,
    H Hpq)

```

-- 8ª demostración

example

```

(H : p ∨ q → r)
: (p → r) ∧ (q → r) :=
and.intro
  ( assume Hp : p,
    H (or.inl Hp))
  ( assume Hq : q,
    H (or.inr Hq))

```

-- 9ª demostración

example

```

(H : p ∨ q → r)
: (p → r) ∧ (q → r) :=
and.intro
  (λ Hp, H (or.inl Hp))
  (λ Hq, H (or.inr Hq))

```

```
-- 10ª demostración
example
  (H : p ∨ q → r)
  : (p → r) ∧ (q → r) :=
  ⟨λ Hp, H (or.inl Hp),
   λ Hq, H (or.inr Hq)⟩
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).





# Capítulo 5

## Ejercicios sobre negaciones

### 5.1. $p \vdash \neg\neg p$

```
-- Ejercicio. Demostrar
--    $p \vdash \neg\neg p$ 
```

```
import tactic
variable (p : Prop)
```

```
-- 1ª demostración
```

```
example
  (H : p)
  :  $\neg\neg p$  :=
begin
  intro H1,
  apply H1 H,
end
```

```
-- 2ª demostración
```

```
example
  (H : p)
  :  $\neg\neg p$  :=
λ H1, H1 H
```

```
-- 3ª demostración
```

```
example
  (H : p)
  :  $\neg\neg p$  :=
-- by library_search
```

```

not_not.mpr H

-- 4ª demostración
example
  (H : p)
  :  $\neg\neg p :=$ 
assume H1 :  $\neg p$ ,
show false,
  from H1 H

-- 5ª demostración
example
  (H : p)
  :  $\neg\neg p :=$ 
-- by hint
by tauto

-- 6ª demostración
example
  (H : p)
  :  $\neg\neg p :=$ 
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.2. $\neg p \vdash p \rightarrow q$

```

-----
-- Ejercicio. Demostrar
--    $\neg p \vdash p \rightarrow q$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (H :  $\neg p$ )
  :  $p \rightarrow q :=$ 
begin
  intro Hp,
  exfalso,
  apply H,

```

```
    exact Hp,
  end

-- 2ª demostración
example
  (H : ¬p)
  : p → q :=
begin
  intro Hp,
  exfalso,
  exact H Hp,
end

-- 3ª demostración
example
  (H : ¬p)
  : p → q :=
begin
  intro Hp,
  exact absurd Hp H,
end

-- 4ª demostración
example
  (H : ¬p)
  : p → q :=
λ Hp, absurd Hp H

-- 5ª demostración
example
  (H : ¬p)
  : p → q :=
-- by library_search
not.elim H

-- 6ª demostración
example
  (H : ¬p)
  : p → q :=
assume Hp : p,
show q,
from absurd Hp H
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 5.3. $p \rightarrow q \vdash \neg q \rightarrow \neg p$

```
-- Ejercicio. Demostrar
--    $p \rightarrow q \vdash \neg q \rightarrow \neg p$ 
```

```
import tactic
variables (p q : Prop)
```

```
-- 1ª demostración
```

```
example
  (H : p → q)
  : ¬q → ¬p :=
```

```
begin
  intro Hnq,
  intro Hp,
  apply Hnq,
  exact H Hp,
```

```
end
```

```
-- 2ª demostración
```

```
example
  (H : p → q)
  : ¬q → ¬p :=
```

```
begin
  intro Hnq,
  intro Hp,
  exact Hnq (H Hp),
```

```
end
```

```
-- 3ª demostración
```

```
example
  (H : p → q)
  : ¬q → ¬p :=
```

```
begin
  intros Hnq Hp,
  exact Hnq (H Hp),
```

```
end
```

```
-- 4ª demostración
```

```
example
  (H : p → q)
  : ¬q → ¬p :=
```

```

λ Hnq Hp, Hnq (H Hp)

-- 5ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
-- by library_search
mt H

-- 6ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
assume Hnq : ¬q,
assume Hp : p,
have Hq : q,
  from H Hp,
show false,
  from Hnq Hq

-- 7ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
assume Hnq : ¬q,
assume Hp : p,
have Hq : q,
  from H Hp,
Hnq Hq

-- 8ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
assume Hnq : ¬q,
assume Hp : p,
Hnq (H Hp)

-- 9ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
assume Hnq : ¬q,
λ Hp, Hnq (H Hp)

```

```

-- 10ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
λ Hnq Hp, Hnq (H Hp)

-- 11ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
-- by hint
by tauto

-- 12ª demostración
example
  (H : p → q)
  : ¬q → ¬p :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.4. $p \vee q, \neg q \vdash p$

```

-----
-- Ejercicio. Demostrar
--    $p \vee q, \neg q \vdash p$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (Hpq : p ∨ q)
  (Hnq : ¬q)
  : p :=
begin
  cases Hpq with Hp Hq,
  { exact Hp, },
  { exact absurd Hq Hnq, },
end

-- 2ª demostración

```

```

example
  (Hpq : p  $\vee$  q)
  (Hnq :  $\neg$ q)
  : p :=
-- by library_search
or.resolve_right Hpq Hnq

-- 3ª demostración
example
  (Hpq : p  $\vee$  q)
  (Hnq :  $\neg$ q)
  : p :=
or.elim Hpq
  ( assume Hp : p,
    show p,
    from Hp)
  ( assume Hq : q,
    show p,
    from absurd Hq Hnq)

-- 4ª demostración
example
  (Hpq : p  $\vee$  q)
  (Hnq :  $\neg$ q)
  : p :=
or.elim Hpq
  ( assume Hp : p,
    show p,
    from Hp)
  ( assume Hq : q,
    absurd Hq Hnq)

-- 5ª demostración
example
  (Hpq : p  $\vee$  q)
  (Hnq :  $\neg$ q)
  : p :=
or.elim Hpq
  ( assume Hp : p,
    show p,
    from Hp)
  (  $\lambda$  Hq, absurd Hq Hnq)

-- 6ª demostración
example

```

```

(Hpq : p ∨ q)
(Hnq : ¬q)
: p :=
or.elim Hpq
  ( assume Hp : p,
    Hp)
  ( λ Hq, absurd Hq Hnq)

-- 7ª demostración
example
  (Hpq : p ∨ q)
  (Hnq : ¬q)
  : p :=
or.elim Hpq id (λ Hq, absurd Hq Hnq)

-- 8ª demostración
example
  (Hpq : p ∨ q)
  (Hnq : ¬q)
  : p :=
-- by hint
by tauto

-- 9ª demostración
example
  (Hpq : p ∨ q)
  (Hnq : ¬q)
  : p :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.5. $p \vee q, \neg p \vdash q$

```

-----
-- Ejercicio. Demostrar
--    $p \vee q, \neg p \vdash q$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración

```



```

example
  (Hpq : p  $\vee$  q)
  (Hnp :  $\neg$ p)
  : q :=
begin
  cases Hpq with Hp Hq,
  { exact absurd Hp Hnp, },
  { exact Hq, },
end

-- 2ª demostración
example
  (Hpq : p  $\vee$  q)
  (Hnp :  $\neg$ p)
  : q :=
or.elim Hpq ( $\lambda$  Hp, absurd Hp Hnp) id

-- 3ª demostración
example
  (Hpq : p  $\vee$  q)
  (Hnp :  $\neg$ p)
  : q :=
-- by library_search
or.resolve_left Hpq Hnp

-- 4ª demostración
example
  (Hpq : p  $\vee$  q)
  (Hnp :  $\neg$ p)
  : q :=
or.elim Hpq
(
  assume Hp : p,
  show q,
  from absurd Hp Hnp
)
(
  assume Hq : q,
  show q,
  from Hq
)

-- 5ª demostración
example
  (Hpq : p  $\vee$  q)
  (Hnp :  $\neg$ p)
  : q :=
-- by hint
by tauto

```

```
-- 6ª demostración
```

```
example
```

```
  (Hpq : p ∨ q)
```

```
  (Hnp: ¬p)
```

```
  : q :=
```

```
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.6. $p \vee q \vdash \neg(\neg p \wedge \neg q)$

```
-- Ejercicio. Demostrar
--    $p \vee q \vdash \neg(\neg p \wedge \neg q)$ 
```

```
import tactic
```

```
variables (p q : Prop)
```

```
-- 1ª demostración
```

```
example
```

```
  (H : p ∨ q)
```

```
  : ¬(¬p ∧ ¬q) :=
```

```
begin
```

```
  intro H1,
```

```
  cases H1 with H2 H3,
```

```
  cases H with H4 H5,
```

```
  { exact H2 H4, },
```

```
  { exact H3 H5, },
```

```
end
```

```
-- 2ª demostración
```

```
example
```

```
  (H : p ∨ q)
```

```
  : ¬(¬p ∧ ¬q) :=
```

```
begin
```

```
  rintro ⟨H2, H3⟩,
```

```
  cases H with H4 H5,
```

```
  { exact H2 H4, },
```

```
  { exact H3 H5, },
```

```
end
```

```
-- 3ª demostración
```

```

example
  (H : p ∨ q)
  : ¬(¬p ∧ ¬q) :=
λ ⟨H2, H3⟩, or.elim H (λ H4, H2 H4) (λ H5, H3 H5)

-- 4ª demostración
example
  (H : p ∨ q)
  : ¬(¬p ∧ ¬q) :=
-- by library_search
or_iff_not_and_not.mp H

-- 5ª demostración
example
  (H : p ∨ q)
  : ¬(¬p ∧ ¬q) :=
assume H3 : ¬p ∧ ¬q,
or.elim H
  ( assume Hp : p,
    show false,
    from absurd Hp (and.left H3))
  ( assume Hq : q,
    show false,
    from absurd Hq (and.right H3))

-- 6ª demostración
example
  (H : p ∨ q)
  : ¬(¬p ∧ ¬q) :=
-- by hint
by tauto

-- 7ª demostración
example
  (H : p ∨ q)
  : ¬(¬p ∧ ¬q) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.7. $p \wedge q \vdash \neg(\neg p \vee \neg q)$

```

-----
-- Ejercicio. Demostrar
--    $p \wedge q \vdash \neg(\neg p \vee \neg q)$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (H : p ∧ q)
  : ¬(¬p ∨ ¬q) :=
begin
  intro H1,
  cases H1 with H2 H3,
  { apply H2,
    exact H.left, },
  { apply H3,
    exact H.right, },
end

-- 2ª demostración
example
  (H : p ∧ q)
  : ¬(¬p ∨ ¬q) :=
begin
  intro H1,
  cases H1 with H2 H3,
  { exact H2 H.left, },
  { exact H3 H.right, },
end

-- 3ª demostración
example
  (H : p ∧ q)
  : ¬(¬p ∨ ¬q) :=
λ H1, or.elim H1 (λ H2, H2 H.1) (λ H3, H3 H.2)

-- 4ª demostración
example
  (H : p ∧ q)
  : ¬(¬p ∨ ¬q) :=

```

```

begin
  rintro (H2 | H3),
  { exact H2 H.left, },
  { exact H3 H.right, },
end

-- 5ª demostración
example
  (H : p ∧ q)
  : ¬(¬p ∨ ¬q) :=
-- by library_search
and_iff_not_or_not.mp H

-- 6ª demostración
example
  (H : p ∧ q)
  : ¬(¬p ∨ ¬q) :=
-- by hint
by tauto

-- 7ª demostración
example
  (H : p ∧ q)
  : ¬(¬p ∨ ¬q) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.8. $\neg(p \vee q) \vdash \neg p \wedge \neg q$

```

-----
-- Ejercicio. Demostrar
--    $\neg(p \vee q) \vdash \neg p \wedge \neg q$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (H : ¬(p ∨ q))
  : ¬p ∧ ¬q :=
begin

```

```

split,
{ intro Hp,
  apply H,
  exact or.inl Hp, },
{ intro Hq,
  apply H,
  exact or.inr Hq, },
end

-- 2ª demostración
example
  (H : ¬(p ∨ q))
  : ¬p ∧ ¬q :=
begin
  split,
  { intro Hp,
    exact H (or.inl Hp), },
  { intro Hq,
    exact H (or.inr Hq), },
end

-- 3ª demostración
example
  (H : ¬(p ∨ q))
  : ¬p ∧ ¬q :=
< λ Hp, H (or.inl Hp),
  λ Hq, H (or.inr Hq) >

-- 4ª demostración
example
  (H : ¬(p ∨ q))
  : ¬p ∧ ¬q :=
-- by library_search
not_or_distrib.mp H

-- 5ª demostración
example
  (H : ¬(p ∨ q))
  : ¬p ∧ ¬q :=
have H1 : ¬p, from
  assume Hp : p,
  have H2: p ∨ q,
    from or.inl Hp,
  show false,
    from absurd H2 H,

```

```

have H3 :  $\neg q$ , from
  assume Hq : q,
  have H4:  $p \vee q$ ,
    from or.inr Hq,
  show false,
    from absurd H4 H,
show  $\neg p \wedge \neg q$ ,
  from and.intro H1 H3

```

```
-- 6ª demostración
```

```

example
  (H :  $\neg(p \vee q)$ )
  :  $\neg p \wedge \neg q :=$ 

```

```
-- by hint
```

```
by tauto
```

```
-- 7ª demostración
```

```

example
  (H :  $\neg(p \vee q)$ )
  :  $\neg p \wedge \neg q :=$ 

```

```
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.9. $\neg p \wedge \neg q \vdash \neg(p \vee q)$

```

-- Ejercicio. Demostrar
--  $\neg p \wedge \neg q \vdash \neg(p \vee q)$ 

```

```

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (H :  $\neg p \wedge \neg q$ )
  :  $\neg(p \vee q) :=$ 
begin
  intro H1,
  cases H1 with H2 H3,
  { exact absurd H2 H.1, },
  { exact absurd H3 H.2, },

```

```

end

-- 2ª demostración
example
  (H : ¬p ∧ ¬q)
  : ¬(p ∨ q) :=
λ H1, or.elim H1 (λ H2, absurd H2 H.1) (λ H3, absurd H3 H.2)

-- 3ª demostración
example
  (H : ¬p ∧ ¬q)
  : ¬(p ∨ q) :=
-- by library_search
not_or_distrib.mpr H

-- 4ª demostración
example
  (H : ¬p ∧ ¬q)
  : ¬(p ∨ q) :=
assume Hpq : p ∨ q,
or.elim Hpq
( assume Hp : p,
  show false,
  from absurd Hp H.left)
( assume Hq : q,
  show false,
  from absurd Hq H.right)

-- 5ª demostración
example
  (H : ¬p ∧ ¬q)
  : ¬(p ∨ q) :=
-- by hint
by tauto

-- 6ª demostración
example
  (H : ¬p ∧ ¬q)
  : ¬(p ∨ q) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).



**5.10.  $\neg p \vee \neg q \vdash \neg(p \wedge q)$** 

```

-----
-- Ejercicio. Demostrar
--    $\neg p \vee \neg q \vdash \neg(p \wedge q)$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example
  (H :  $\neg p \vee \neg q$ )
  :  $\neg(p \wedge q)$  :=
begin
  intro Hpq,
  cases H with Hnp Hnq,
  { apply Hnp,
    exact Hpq.left, },
  { apply Hnq,
    exact Hpq.right, },
end

-- 2ª demostración
example
  (H :  $\neg p \vee \neg q$ )
  :  $\neg(p \wedge q)$  :=
begin
  intro Hpq,
  cases H with Hnp Hnq,
  { exact Hnp Hpq.1, },
  { exact Hnq Hpq.2, },
end

-- 3ª demostración
example
  (H :  $\neg p \vee \neg q$ )
  :  $\neg(p \wedge q)$  :=
begin
  intro Hpq,
  exact or.elim H ( $\lambda$  Hnp, Hnp Hpq.1) ( $\lambda$  Hnq, Hnq Hpq.2),
end

-- 4ª demostración

```

```

example
  (H : ¬p ∨ ¬q)
  : ¬(p ∧ q) :=
λ Hpq, or.elim H (λ Hnp, Hnp Hpq.1) (λ Hnq, Hnq Hpq.2)

-- 5ª demostración
example
  (H : ¬p ∨ ¬q)
  : ¬(p ∧ q) :=
-- by library_search
not_and_distrib.mpr H

-- 6ª demostración
example
  (H : ¬p ∨ ¬q)
  : ¬(p ∧ q) :=
assume Hpq : p ∧ q,
or.elim H
  ( assume Hnp : ¬p,
    show false,
    from Hnp (and.left Hpq))
  ( assume Hnq : ¬q,
    show false,
    from Hnq (and.right Hpq))

-- 7ª demostración
example
  (H : ¬p ∨ ¬q)
  : ¬(p ∧ q) :=
-- by hint
by tauto

-- 8ª demostración
example
  (H : ¬p ∨ ¬q)
  : ¬(p ∧ q) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

**5.11.  $\neg(p \wedge \neg p)$** 

```

-----
-- Ejercicio. Demostrar
--    $\vdash \neg(p \wedge \neg p)$ 
-----

import tactic
variables (p q : Prop)

-- 1ª demostración
example :
   $\neg(p \wedge \neg p) :=$ 
begin
  intro H,
  apply H.right,
  exact H.left,
end

-- 2ª demostración
example :
   $\neg(p \wedge \neg p) :=$ 
begin
  intro H,
  exact H.right (H.left),
end

-- 3ª demostración
example :
   $\neg(p \wedge \neg p) :=$ 
 $\lambda H, H.right (H.left)$ 

-- 4ª demostración
example :
   $\neg(p \wedge \neg p) :=$ 
begin
  rintro ⟨H1, H2⟩,
  exact H2 H1,
end

-- 5ª demostración
example :
   $\neg(p \wedge \neg p) :=$ 
 $\lambda \langle H1, H2 \rangle, H2 H1$ 

```

```

-- 6ª demostración
example :
  ¬(p ∧ ¬p) :=
-- by suggest
(and_not_self p).mp

-- 7ª demostración
example :
  ¬(p ∧ ¬p) :=
assume H : p ∧ ¬p,
have H1 : p,
  from and.left H,
have H2 : ¬p,
  from and.right H,
show false,
  from H2 H1

-- 8ª demostración
example :
  ¬(p ∧ ¬p) :=
-- by hint
by tauto

-- 9ª demostración
example :
  ¬(p ∧ ¬p) :=
by finish

-- 10ª demostración
example :
  ¬(p ∧ ¬p) :=
by simp

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 5.12. $p \wedge \neg p \vdash q$

```

-----
-- Ejercicio. Demostrar
--    $p \wedge \neg p \vdash q$ 
-----

import tactic

```

```
variables (p q : Prop)

-- 1ª demostración
example
  (H : p ∧ ¬p)
  : q :=
begin
  exfalso,
  apply H.2,
  exact H.1,
end

-- 2ª demostración
example
  (H : p ∧ ¬p)
  : q :=
begin
  exfalso,
  exact H.2 H.1,
end

-- 3ª demostración
example
  (H : p ∧ ¬p)
  : q :=
false.elim (H.2 H.1)

-- 4ª demostración
example
  (H : p ∧ ¬p)
  : q :=
have Hp : p,
  from and.left H,
have Hnp : ¬p,
  from and.right H,
have Hf : false,
  from Hnp Hp,
show q,
  from false.elim Hf

-- 5ª demostración
example
  (H : p ∧ ¬p)
  : q :=
-- by hint
```

```
by tauto

-- 6ª demostración
example
  (H : p ∧ ¬p)
  : q :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

# Capítulo 6

## Ejercicios de lógica clásica

### 6.1. $\neg\neg p \vdash p$

```
-- Ejercicio . Demostrar
--       $\neg\neg p \vdash p$ 
```

```
import tactic
variable (p : Prop)

open_locale classical
```

```
-- 1ª demostración
```

```
example
  (h1 :  $\neg\neg p$ )
  : p :=
by_contra
  ( assume h2 :  $\neg p$ ,
    show false,
    from h1 h2 )
```

```
-- 2ª demostración
```

```
example
  (h1 :  $\neg\neg p$ )
  : p :=
by_contra
  ( assume h2 :  $\neg p$ ,
    h1 h2 )
```

```
-- 3ª demostración
```

```

example
  (h1 : ¬¬p)
  : p :=
by_contra (λ h2, h1 h2)

-- 4ª demostración
example
  (h1 : ¬¬p)
  : p :=
-- by library_search
not_not.mp h1

-- 5ª demostración
example
  (h1 : ¬¬p)
  : p :=
begin
  by_contradiction h2,
  exact h1 h2,
end

-- 6ª demostración
example
  (h1 : ¬¬p)
  : p :=
-- by hint
by tauto

-- 7ª demostración
lemma aux
  (h1 : ¬¬p)
  : p :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 6.2. $\vdash p \vee \neg p$

```

-----
-- Ejercicio. Demostrar
--    $\vdash p \vee \neg p$ 
-----

```



```
import tactic
variable (p : Prop)

open_locale classical

-- 1ª demostración
example : p ∨ ¬p :=
by_contradiction
( assume h1 : ¬(p ∨ ¬p),
  have h2 : ¬p, from
    assume h3 : p,
    have h4 : p ∨ ¬p, from or.inl h3,
    show false, from h1 h4,
  have h5 : p ∨ ¬p, from or.inr h2,
  show false, from h1 h5 )

-- 2ª demostración
example : p ∨ ¬p :=
by_contradiction
( assume h1 : ¬(p ∨ ¬p),
  have h2 : ¬p, from
    assume h3 : p,
    have h4 : p ∨ ¬p, from or.inl h3,
    show false, from h1 h4,
  have h5 : p ∨ ¬p, from or.inr h2,
  h1 h5 )

-- 3ª demostración
example : p ∨ ¬p :=
by_contradiction
( assume h1 : ¬(p ∨ ¬p),
  have h2 : ¬p, from
    assume h3 : p,
    have h4 : p ∨ ¬p, from or.inl h3,
    show false, from h1 h4,
  h1 (or.inr h2) )

-- 4ª demostración
example : p ∨ ¬p :=
by_contradiction
( assume h1 : ¬(p ∨ ¬p),
  have h2 : ¬p, from
    assume h3 : p,
    have h4 : p ∨ ¬p, from or.inl h3,
    h1 h4,
```

```

    h1 (or.inr h2) )

-- 5ª demostración
example : p ∨ ¬p :=
by_contradiction
  ( assume h1 : ¬(p ∨ ¬p),
    have h2 : ¬p, from
      assume h3 : p,
        h1 (or.inl h3),
    h1 (or.inr h2) )

-- 6ª demostración
example : p ∨ ¬p :=
by_contradiction
  ( assume h1 : ¬(p ∨ ¬p),
    have h2 : ¬p, from
      λ h3, h1 (or.inl h3),
    h1 (or.inr h2) )

-- 7ª demostración
example : p ∨ ¬p :=
by_contradiction
  ( assume h1 : ¬(p ∨ ¬p),
    h1 (or.inr (λ h3, h1 (or.inl h3))) )

-- 8ª demostración
example : p ∨ ¬p :=
by_contradiction
  ( λ h1, h1 (or.inr (λ h3, h1 (or.inl h3))) )

-- 9ª demostración
example : p ∨ ¬p :=
-- by library_search
em p

-- #print axioms em

-- 10ª demostración
example : p ∨ ¬p :=
begin
  by_contra h1,
  apply h1,
  apply or.inr,
  intro h2,
  apply h1,

```

```
    exact or.inl h2,
end

-- 11ª demostración
example : p  $\vee$   $\neg$ p :=
begin
  by_contra h1,
  apply h1,
  apply or.inr,
  intro h2,
  exact h1 (or.inl h2),
end

-- 12ª demostración
example : p  $\vee$   $\neg$ p :=
begin
  by_contra h1,
  apply h1,
  apply or.inr,
  exact  $\lambda$  h2, h1 (or.inl h2),
end

-- 13ª demostración
example : p  $\vee$   $\neg$ p :=
begin
  by_contra h1,
  apply h1,
  exact or.inr ( $\lambda$  h2, h1 (or.inl h2)),
end

-- 14ª demostración
example : p  $\vee$   $\neg$ p :=
begin
  by_contra h1,
  exact h1 (or.inr ( $\lambda$  h2, h1 (or.inl h2))),
end

-- 15ª demostración
example : p  $\vee$   $\neg$ p :=
by_contra ( $\lambda$  h1, h1 (or.inr ( $\lambda$  h2, h1 (or.inl h2))))

-- 16ª demostración
example : p  $\vee$   $\neg$ p :=
begin
  by_contra h1,
```

```

    apply h1,
    right,
    intro h2,
    apply h1,
    left,
    exact h2,
end

-- 17ª demostración
example : p ∨ ¬p :=
-- by hint
by tauto

-- 18ª demostración
example : p ∨ ¬p :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

### 6.3. $((p \rightarrow q) \rightarrow p) \rightarrow p$

```

-----
-- Ejercicio. Demostrar
--   ⊢ ((p → q) → p) → p
-----

import tactic
variables (p q : Prop)

open_locale classical

-- 1ª demostración
example :
  ((p → q) → p) → p :=
begin
  intro h1,
  by_cases h2 : p → q,
  { exact h1 h2, },
  { by_contra h3,
    apply h2,
    intro h4,
    exfalso,
    exact h3 h4, },

```

```

end

-- 2ª demostración
example :
  ((p → q) → p) → p :=
begin
  by_cases hp : p,
  { intro h1,
    exact hp, },
  { intro h2,
    exact h2 hp.elim, },
end

-- 3ª demostración
example :
  ((p → q) → p) → p :=
if hp : p then λ h, hp else λ h, h hp.elim

-- 4ª demostración
example :
  ((p → q) → p) → p :=
-- by library_search
peirce p q

-- 5ª demostración
example :
  ((p → q) → p) → p :=
assume h1 : (p → q) → p,
show p, from
  by_contradiction
  ( assume h2 : ¬p,
    have h3 : ¬(p → q),
      by exact mt h1 h2,
    have h4 : p → q, from
      assume h5 : p,
      show q,
      from not.elim h2 h5,
    show false,
    from h3 h4)

-- 6ª demostración
example :
  ((p → q) → p) → p :=
-- by hint
by tauto

```

```
-- 7ª demostración
example :
  ((p → q) → p) → p :=
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 6.4. $\neg q \rightarrow \neg p \vdash p \rightarrow q$

```
-- Ejercicio. Demostrar
--    $\neg q \rightarrow \neg p \vdash p \rightarrow q$ 
--
```

```
import tactic
variables (p q : Prop)
```

```
open_locale classical
```

```
-- 1ª demostración
```

```
example
  (H :  $\neg q \rightarrow \neg p$ )
  :  $p \rightarrow q$  :=
begin
  intro Hp,
  by_contra Hnq,
  apply not.elim _ Hp,
  exact H Hnq,
end
```

```
-- 2ª demostración
```

```
example
  (H :  $\neg q \rightarrow \neg p$ )
  :  $p \rightarrow q$  :=
-- by library_search
not_imp_not.mp H
```

```
-- 3ª demostración
```

```
example
  (H :  $\neg q \rightarrow \neg p$ )
  :  $p \rightarrow q$  :=
assume Hp : p,
show q, from
```

```

by_contradiction
  ( assume Hnq : ¬q,
    have Hnp : ¬p,
      from H Hnq,
    show false,
      from Hnp Hp )

-- 4ª demostración
example
  (H : ¬q → ¬p)
  : p → q :=
-- by hint
by tauto

-- 5ª demostración
example
  (H : ¬q → ¬p)
  : p → q :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 6.5. $\neg(\neg p \wedge \neg q) \vdash p \vee q$

```

-----
-- Ejercicio. Demostrar
--    $\neg(\neg p \wedge \neg q) \vdash p \vee q$ 
-----

import tactic
variables (p q r : Prop)

open_locale classical

-- 1ª demostración
example
  (H : ¬(¬p ∧ ¬q))
  : p ∨ q :=
begin
  by_cases Hp : p,
  { exact or.inl Hp, },
  { by_cases Hq : q,
    { exact or.inr Hq, },

```

```

    { exfalso,
      apply H,
      exact and.intro Hp Hq, }},
end

-- 2ª demostración
example
  (H : ¬(¬p ∧ ¬q))
  : p ∨ q :=
-- by library_search
or_iff_not_and_not.mpr H

-- 3ª demostración
example
  (H : ¬(¬p ∧ ¬q))
  : p ∨ q :=
or.elim (em p)
  ( assume Hp : p,
    show p ∨ q,
    from or.inl Hp)
  ( assume Hnp : ¬p,
    show p ∨ q, from
      or.elim (em q)
        ( assume Hq : q,
          show p ∨ q,
          from or.inr Hq)
        ( assume Hnq : ¬q,
          have H' : ¬p ∧ ¬q,
            from and.intro Hnp Hnq,
          show p ∨ q,
          from not.elim H H'))

-- 4ª demostración
example
  (H : ¬(¬p ∧ ¬q))
  : p ∨ q :=
or.elim (em p)
  or.inl
  (λ Hnp, or.elim (em q)
    or.inr
    (λ Hnq, not.elim H (and.intro Hnp Hnq)))

-- 5ª demostración
example
  (H : ¬(¬p ∧ ¬q))

```



```

: p ∨ q :=
-- by hint
by tauto

-- 6ª demostración
example
  (H : ¬(¬p ∧ ¬q))
  : p ∨ q :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 6.6. $\neg(\neg p \vee \neg q) \vdash p \wedge q$

```

-----
-- Ejercicio. Demostrar
--   ¬(¬p ∨ ¬q) ⊢ p ∧ q
-----

```

```

import tactic
variables (p q : Prop)

open_locale classical

-- 1ª demostración
example
  (H : ¬(¬p ∨ ¬q))
  : p ∧ q :=
begin
  split,
  { by_contra Hnp,
    apply H,
    exact or.inl Hnp, },
  { by_contra Hnq,
    apply H,
    exact or.inr Hnq, },
end

-- 2ª demostración
example
  (H : ¬(¬p ∨ ¬q))
  : p ∧ q :=
begin

```

```

split,
{ by_contra Hnp,
  exact H (or.inl Hnp), },
{ by_contra Hnq,
  exact H (or.inr Hnq), },
end

-- 3ª demostración
example
(H : ¬(¬p ∨ ¬q))
: p ∧ q :=
begin
  split,
  { exact by_contra (λ Hnp, H (or.inl Hnp)), },
  { exact by_contra (λ Hnq, H (or.inr Hnq)), },
end

-- 4ª demostración
example
(H : ¬(¬p ∨ ¬q))
: p ∧ q :=
⟨by_contra (λ Hnp, H (or.inl Hnp)),
 by_contra (λ Hnq, H (or.inr Hnq))⟩

-- 5ª demostración
example
(H : ¬(¬p ∨ ¬q))
: p ∧ q :=
-- by library_search
and_iff_not_or_not.mpr H

-- 6ª demostración
example
(H : ¬(¬p ∨ ¬q))
: p ∧ q :=
and.intro
( show p, from by_contradiction
  ( assume Hnp : ¬p,
    have H' : ¬p ∨ ¬q,
      from or.inl Hnp,
    show false,
      from H H'))
( show q, from by_contradiction
  ( assume Hnq : ¬q,
    have H' : ¬p ∨ ¬q,

```

```

        from or.inr Hnq,
        show false,
        from H H'))

-- 7ª demostración
example
  (H :  $\neg(\neg p \vee \neg q)$ )
  :  $p \wedge q :=$ 
-- by hint
by tauto

-- 8ª demostración
example
  (H :  $\neg(\neg p \vee \neg q)$ )
  :  $p \wedge q :=$ 
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 6.7. $\neg(p \wedge q) \vdash \neg p \vee \neg q$

```

-----
-- Ejercicio. Demostrar
--    $\neg(p \wedge q) \vdash \neg p \vee \neg q$ 
-----

import tactic
variables (p q r : Prop)

open_locale classical

-- 1ª demostración
example
  (H :  $\neg(p \wedge q)$ )
  :  $\neg p \vee \neg q :=$ 
begin
  by_cases Hp : p,
  { by_cases Hq : q,
    { ex falso,
      apply H,
      exact ⟨Hp, Hq⟩, },
    { exact or.inr Hq, }},
  { exact or.inl Hp, },

```

```

end

-- 2ª demostración
example
  (H : ¬(p ∧ q))
  : ¬p ∨ ¬q :=
if Hp : p
then if Hq : q
  then not.elim H ⟨Hp, Hq⟩
  else or.inr Hq
else or.inl Hp

-- 3ª demostración
example
  (H : ¬(p ∧ q))
  : ¬p ∨ ¬q :=
-- by library_search
not_and_distrib.mp H

-- 4ª demostración
example
  (H : ¬(p ∧ q))
  : ¬p ∨ ¬q :=
or.elim (em p)
( assume Hp : p,
  or.elim (em q)
  ( assume Hq : q,
    show ¬p ∨ ¬q,
    from not.elim H ⟨Hp, Hq⟩)
  ( assume Hnq : ¬q,
    show ¬p ∨ ¬q,
    from or.inr Hnq))
( assume Hnp : ¬p,
  show ¬p ∨ ¬q,
  from or.inl Hnp)

-- 5ª demostración
example
  (H : ¬(p ∧ q))
  : ¬p ∨ ¬q :=
or.elim (em p)
(λ Hp, or.elim (em q)
  (λ Hq, not.elim H ⟨Hp, Hq⟩)
  or.inr)
or.inl

```

```
-- 6ª demostración
```

```
example
```

```
  (H :  $\neg(p \wedge q)$ )
```

```
  :  $\neg p \vee \neg q :=$ 
```

```
-- by hint
```

```
by tauto
```

```
-- 7ª demostración
```

```
example
```

```
  (H :  $\neg(p \wedge q)$ )
```

```
  :  $\neg p \vee \neg q :=$ 
```

```
by finish
```

Enlaces al [código](#) y a la [sesión en Lean Web](#).

## 6.8. $\vdash (p \rightarrow q) \vee (q \rightarrow p)$

```
-- -----  
-- Ejercicio. Demostrar  
--  $\vdash (p \rightarrow q) \vee (q \rightarrow p)$   
-- -----
```

```
import tactic
```

```
variables (p q : Prop)
```

```
open_locale classical
```

```
-- 1ª demostración
```

```
example :
```

```
  ( $p \rightarrow q$ )  $\vee$  ( $q \rightarrow p$ ) :=
```

```
begin
```

```
  by_cases H1 : p,
```

```
  { right,
```

```
    intro,
```

```
    exact H1, },
```

```
  { left,
```

```
    intro H2,
```

```
    exfalso,
```

```
    exact H1 H2, },
```

```
end
```

```
-- 2ª demostración
```

```
example :
```

```

(p → q) ∨ (q → p) :=
begin
  cases (em p) with Hp Hnp,
  { exact or.inr (λ Hq, Hp), },
  { exact or.inl (λ Hp, not.elim Hnp Hp), },
end

-- 3ª demostración
example :
  (p → q) ∨ (q → p) :=
or.elim (em p)
  (λ Hp, or.inr (λ Hq, Hp))
  (λ Hnp, or.inl (λ Hp, not.elim Hnp Hp))

-- 4ª demostración
example :
  (p → q) ∨ (q → p) :=
if Hp : p
  then or.inr (λ _, Hp)
  else or.inl (λ H, not.elim Hp H)

-- 5ª demostración
example :
  (p → q) ∨ (q → p) :=
-- by hint
by tauto

-- 6ª demostración
example :
  (p → q) ∨ (q → p) :=
by finish

```

Enlaces al [código](#) y a la [sesión en Lean Web](#)

# Capítulo 7

## Bibliografía

- [Deducción natural proposicional](#). ~ J.A. Alonso, A. Cordon, M.J. Hidalgo.
- [Lógica con Lean](#) ~ J.A. Alonso.
  - Cap. 2: Lógica proposicional.
- [Logic and proof](#). ~ J. Avigad, R.Y. Lewis, F. van Doorn.
  - Cap. 4: Propositional Logic in Lean.
- [Logic in Computer Science](#). ~ M. Huth, M. Ryan.
  - Cap. 1.2: Propositional logic. Natural deduction.
- [Theorem proving in Lean](#). ~ J. Avigad, L. de Moura, S. Kong.
  - Cap. 3: Propositions and proofs.