4.1

1 a Pedro trabaja en CSA.
`empleado(Pedro,CSA)`

b Juan es amigo de un trabajador de CSA.
$(\exists x \mid empleado(x,CSA) : amigo(Juan,x))$

c Todo trabajador de CSA es arquitecto o ingeniero.
$(\forall x \mid empleado(x,CSA) : arq.x \vee ing.x)$

d Hay trabajadores de CSA que no son amigos.
$(\exists x,y \mid empleado(x,CSA) \wedge empleado(y,CSA) : \neg amigo(x,y))$

e Hay trabajadores de CSA que no son amigos, pero cuando son amigos, ambos son arquitectos o ambos son ingenieros.
$(\exists x,y \mid empleado(x,CSA) \wedge empleado(y,CSA) : \neg amigo(x,y)) \wedge$
$(\forall x,y \mid empleado(x,CSA) \wedge empleado(y,CSA) \wedge amigo(x,y):$
$(arq.x \wedge arq.y) \vee (ing.x \wedge ing.y))$

f Hay trabajadores de CSA que no son amigos de su jefe.
$(\exists x \mid empleado(x,CSA) : \neg amigo(x,jefe.x))$

g En ninguna empresa hay jefes que son amigos de los empleados.
$\neg(\exists e|:(\exists x \mid empleado(x,e) : \neg amigo(x,jefe.x))$

2 En cada caso decidir si x, y, z son variables libres o acotadas en la fórmula.

La siguiente tabla resume las respuestas:

	x libre	x acotada	y libre	y acotada	z libre	z acotada
a	no	sí	no	no	no	no
b	no	sí	no	sí	no	sí
c	sí	no	no	sí	no	sí
d	sí	sí	sí	sí	no	no

4.2

1 Contraejemplo: $P \equiv false$, $R \equiv false$.

2 Contraejemplo: $R \equiv x=0 \vee x=1,\ P \equiv x>0,\ Q \equiv x>1$.

3

Teo: $(\forall x| Q\vee R : P) \Rightarrow (\forall x| Q : P)$
Dem:

$(\forall x| Q\vee R : P)$
$=$ $\langle$ Trueque $\rangle$
$(\forall x|: Q \vee R \Rightarrow P)$
$=$ $\langle$ Definición $\Rightarrow$ $\rangle$
$(\forall x|: \neg(Q \vee R) \vee P)$
$=$ $\langle$ De Morgan $\rangle$
$(\forall x|: (\neg Q \wedge \neg R) \vee P)$

$=$ ⟨ Distr $\vee/\wedge$ ⟩
$(\forall x|: (\neg Q \vee P) \wedge (\neg R \vee P))$
$\Rightarrow$ ⟨ Debilitamiento de cuerpo ⟩
$(\forall x|: \neg Q \vee P)$
$=$ ⟨ Definición $\Rightarrow$ ⟩
$(\forall x|: Q \Rightarrow P)$
$=$ ⟨Trueque⟩
$(\forall x| Q : P)$

[]

4 En la distributividad $\wedge/\exists$ no es necesario garantizar que el rango no sea vacío porque, en ese caso, ambos lados de la equivalencia tienen el valor `false`. En la distributividad $\vee/\exists$, si no se garantiza que el rango sea no vacío en la parte derecha, se podría tener, por ejemplo, $P \equiv true$ y, en ese caso, el lado izquierdo de la equivalencia valdría `true`, mientras que el derecho sería equivalente a `false`.**5**

Teo: $(\exists x| Q \vee R : P) \Leftarrow (\exists x| Q : P)$

Dem:

$(\exists x| Q \vee R : P)$
$=$ ⟨Trueque⟩
$(\exists x|: (Q \vee R) \wedge P)$
$=$ ⟨Distribución $\wedge/\vee$⟩
$(\exists x|: (Q \wedge P) \vee (R \wedge P))$
$\Leftarrow$ ⟨Debilitamiento de cuerpo⟩
$(\exists x|: Q \wedge P)$
$=$ ⟨Trueque⟩
$(\exists x| Q : P)$

[]

Ejercicios 4.3

1

a $(\forall x| per.x : con.x)$, donde:
- $per.x \approx$ "x es persona"
- $con.x \approx$ "x puede conducir carro"

b $(\forall x| hum.x : hom.x \vee muj.x)$, donde:
- $hum.x \approx$ "x es humano"
- $hom.x \approx$ "x es hombre"
- $muj.x \approx$ "x es mujer"

c $(\forall x{:}int|: (\exists y{:}int|: div(y,x)))$, donde:
- $div(y,x) \approx$ "y divide a x"

d $\neg(\forall p{:}Perro |: perezoso.p) \Rightarrow (\exists p{:}Perro|: \neg perezoso.p)$, donde:
- $perezoso.x \approx$ "x es perezoso"

e $(\exists y| bac.y : \neg uni.y)$, donde:
- $bac.x \approx$ "x termina bachillerato"
- $uni.x \approx$ "x va a la universidad"

2 Modelaje:

```
a.x     ≈ "x es un abra"
u(x,y)≈ "cadabra resultado de unir la cadabra x con la cadabra y"
```

Hechos:

```
[h0] (∀x,y|: (∃z|: unir(x,y) = z))
[h1] (∀x| a.x : (∀y|: u(x,y) = y ∧ u(y,x) = y))
[h2] a.p
[h3] a.q
[t]  p = q
```

A demostrar: `h0 ∧h1 ∧ h2 ∧ h3 ⇒ t`

Dem:

Lema 1: `u(p,q)=q ∧ u(q,p)=q`

Dem:

```
   Hip: h0, h1, h2, h3
       true
   =        ⟨Hip h1: (∀x| a.x : (∀y|: u(x,y) = y ∧ u(y,x) = y))⟩
       (∀x| a.x : (∀y|: u(x,y) = y ∧ u(y,x) = y))
   =        ⟨∀-Trueque⟩
       (∀x|: a.x ⇒ (∀y|: u(x,y) = y ∧ u(y,x) = y))
   ⇒        ⟨Instanciación: x:= p⟩
       a.p ⇒ (∀y|: u(p,y)=y ∧ u(y,p)=y)
   ⇒        ⟨Hip h2; Modus Ponens⟩
       (∀y|: u(p,y)=y ∧ u(y,p)=y)
   ⇒        ⟨Instanciación: y:= q⟩
       u(p,q)=q ∧ u(q,p)=q
```

Lema 2: `u(q,p)=p ∧ u(p,q)=p`

Dem:

Cambiando los papeles de `p` y `q` en la demostración del Lema 1.

Ahora:

Teo: `h0 ∧h1 ∧ h2 ∧ h3 ⇒ t`

Dem: Con hipótesis.

```
Hip: h0, h1, h2, h3, Lema 1, Lema 2
   true
=     ⟨Lema1, Lema2, true ∧ true ≡ true⟩
   u(p,q)=q ∧ u(q,p)=q ∧ u(q,p)=p ∧ u(p,q)=p
⇒     ⟨transitividad⟩
   p=q
```

[]

3

Modelaje:

```
c.x≈   "x clasifica al Mundial de Fútbol de Suráfrica"
Bra           ≈ "Brasil"
Arg           ≈ "Argentina"
Col           ≈ "Colombia"
Bol           ≈ "Bolivia"
```

$g(x,y)$ ≈ "x gana a y"
$p(x,y)$ ≈ "x pierde con y"
$e(x,y)$ ≈ "x empata con y"

Hechos:
[h1] $(\forall x,y|: g(x,y) \equiv \neg p(x,y) \wedge \neg e(x,y))$
[h2] $(\forall x,y|: p(x,y) \equiv \neg g(x,y) \wedge \neg e(x,y))$
[h3] $(\forall x,y|: e(x,y) \equiv \neg g(x,y) \wedge \neg p(x,y))$
[h4] $(\forall x|\ e(x,Bra): g(x,Arg) \vee \neg p(x,Bol) \Rightarrow c.x)$
[h5] $e(Col,Bra)$
[h6] $e(Col,Bol)$
[t] $c.Col$

El teorema a demostrar es:

$$h1 \wedge h2 \wedge h3 \wedge h4 \wedge h5 \wedge h6 \Rightarrow t$$

Dem:

Lema A: $h2 \Rightarrow (\forall x,y|: \neg p(x,y) \equiv g(x,y) \vee e(x,y))$
Dem:
Hip: h2
```
   true
=     ⟨Hip h2⟩
   (∀x,y|: p(x,y) ≡ ¬g(x,y) ∧ ¬e(x,y))
      ⟨ x ≡ y ≡ ¬x ≡ ¬y ⟩
=  (∀x,y|: ¬p(x,y) ≡ ¬(¬g(x,y) ∧ ¬e(x,y)))
      ⟨De Morgan, doble negación⟩
=  (∀x,y|: ¬p(x,y) ≡ g(x,y) ∨ e(x,y))
```

Lema B: $h2 \wedge h6 \Rightarrow \neg p(Col,Bol)$
Dem:
Hip: h2,h6,Lema A
```
   true
=     ⟨Hip: h2, Lema A⟩
   (∀x,y|: ¬p(x,y) ≡ g(x,y) ∨ e(x,y))
=     ⟨Instanciación: x,y:= Col,Bol⟩
   ¬p(Col,Bol) ≡ g(Col,Bol) ∨ e(Col,Bol)
=     ⟨Hip: h6⟩
   ¬p(Col,Bol) ≡ g(Col,Bol) ∨ true
=     ⟨ u ∨ true ≡ true⟩
   ¬p(Col,Bol) ≡ true
=     ⟨≡ identidad⟩
   ¬p(Col,Bol)
```

Hip: h1,...,h6,Lema B
```
   true
=     ⟨Hip h4⟩
   (∀x| e(x,Bra): g(x,Arg) ∨ ¬p(x,Bol) ⇒ c.x)
=     ⟨∀-trueque⟩
   (∀x|: e(x,Bra) ⇒ (g(x,Arg) ∨ ¬p(x,Bol) ⇒ c.x))
=     ⟨∀-Instanciación, x:= Col⟩
```

```
   e(Col,Bra) ⇒ (g(Col,Arg) ∨ ¬p(Col,Bol) ⇒ c. Col)
⇒     ⟨Hip h5: e(Col,Bra), Modus Ponens⟩
   g(Col,Arg) ∨ ¬p(Col,Bol) ⇒ c. Col
=     ⟨Lema B⟩
   g(Col,Arg) ∨ true ⇒ c. Col
=     ⟨ u ∨ true ≡ true⟩
   true ⇒ c.Col
=     ⟨true ⇒ u ≡ u⟩
   c.Col
```

[]