Sección 3.1: 2(b,d,f,g), 3(a,c,e,g), 5

Imagen que contiene Word

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. F(p ≡ r)
2. F(p) ≡ F(r) sustitución caso ≡
3. ((p ≡ q) ≡ (false)) p ϵ {p,q,r} y r ϵ {p,q,r}

Icono

Descripción generada automáticamente

1. F(( p Ʌ q ) V ((¬p)) Ʌ (¬q)))
2. F(p Ʌ q) V F ((¬p)) Ʌ (¬q)) Sustitución textual caso V
3. F(p) Ʌ F(q) V (¬F(p)) Ʌ (¬F(q)) Sustitución textual caso Ʌ y caso V
4. ((p ≡ q) Ʌ (r → s)) V ((¬(p ≡ q)) Ʌ (¬(r → s))) (p,q,r) ϵ {p,q,r}

Un dibujo de una persona

Descripción generada automáticamente con confianza media

1. F((p V r) ← (p Ʌq))
2. F(p ∨ r) ← F (p Ʌ q) Sustitución caso ←
3. (F(p) V F(r)) ← (F(p) Ʌ F(q)) Sustitución caso V y Ʌ
4. (((p ≡ q) ∨ false) ← ((p ≡ q) Ʌ (r → s)) p,q,r ϵ {p,q,r}



F((¬ ((r Ʌ (r ← (p V s))) ≡ (¬((p → q) V (r Ʌ (¬r)))))

F((¬ ((r Ʌ (r ← (p V s))) ≡ F(¬((p → q) V (r Ʌ (¬r))))) Sustitución textual caso ≡

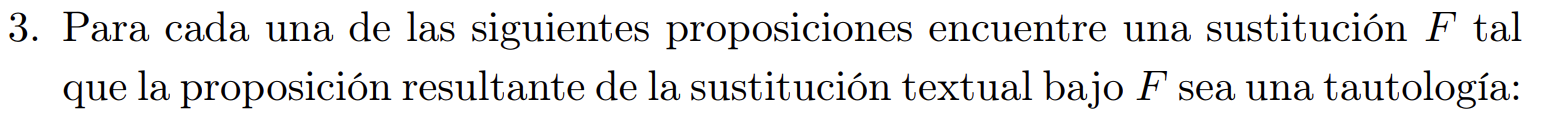
(¬(F(((r Ʌ (r ← (p V s)))) ≡ (¬F(((p → q) V (r Ʌ (¬r))))) ST caso ¬

(¬(F(r)) Ʌ F(r ← (p V s)) ≡ (¬F(p →q)) V (F(r) Ʌ (¬F(r)) ST caso V y caso Ʌ

(¬false) Ʌ F(r) ← F(p V s) ≡ (¬(F(p) → F(q)) V ((false) Ʌ (¬false)) ST caso → y ←, r ϵ {p,q,r}

(¬false) Ʌ (false) ← F(p) V F(s) ≡ (¬(p ≡ q)) → (r → s) V ((false) Ʌ (¬false)) ST caso Ʌ

((¬false) Ʌ ((false) ← ((p ≡ q)) V s))) ≡ ((¬(p ≡ q)) → (r → s)) V (¬false)) (p,q,r) ϵ {p,q,r} Ʌ s {p,q,r}



Dibujo de una persona

Descripción generada automáticamente con confianza media

{p ⟼ (p ≡ p)}

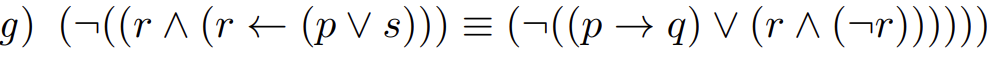
Un dibujo de una persona

Descripción generada automáticamente con confianza baja

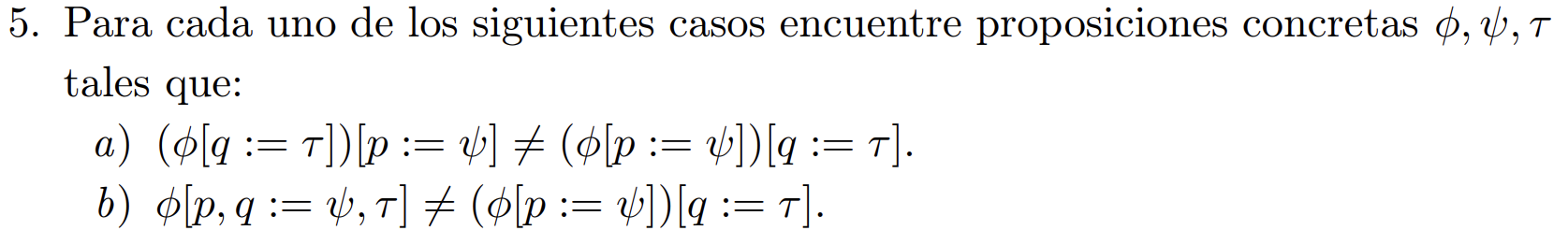
{r ⟼ (p → p)}



{p ⟼ (q ≡ (¬q))}



{r ⟼ (p Ʌ (¬p)), p ⟼ (q ≡ q), q ⟼ (r ≡ (¬r))}



1. ɸ = ((¬q) Ʌ (¬p))

{τ ⟼ (q → p), ψ ⟼ (p → q)}

Primera parte

ɸ = ((¬τ) Ʌ (¬p))

ɸ = ((¬(q → p)) Ʌ (¬p))

ɸ = ((¬(q → ψ)) Ʌ (¬ψ))

ɸ = ((¬(q → (p → q))) Ʌ (¬(p → q)))

Segunda parte

ɸ = ((¬q) Ʌ (¬p))

ɸ = ((¬q) Ʌ (¬ψ))

ɸ = ((¬q) Ʌ (¬(p → q)))

ɸ = ((¬τ) Ʌ (¬(p → τ)))

ɸ = ((¬(q → p)) Ʌ (¬(p → (q → p))))

1. ɸ = ((¬q) Ʌ (¬p))

{τ ⟼ (q → p), ψ ⟼ (p → q)}

Primera parte

ɸ = ((¬q) Ʌ (¬p))

ɸ = ((¬ψ) Ʌ (¬τ))

ɸ = ((¬(p → q)) Ʌ (¬(q → p)))

Segunda parte

ɸ = ((¬q) Ʌ (¬p))

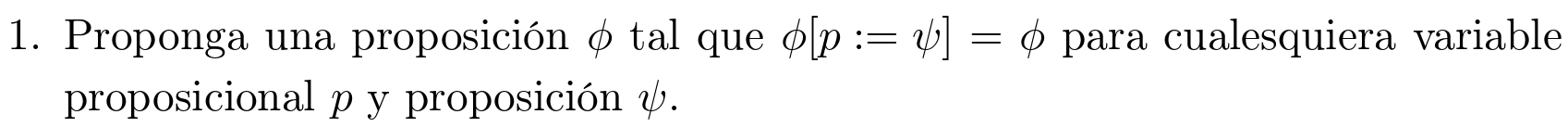
ɸ = ((¬q) Ʌ (¬τ))

ɸ = ((¬q) Ʌ (¬(q → p)))

ɸ = ((¬τ) Ʌ (¬(τ → p)))

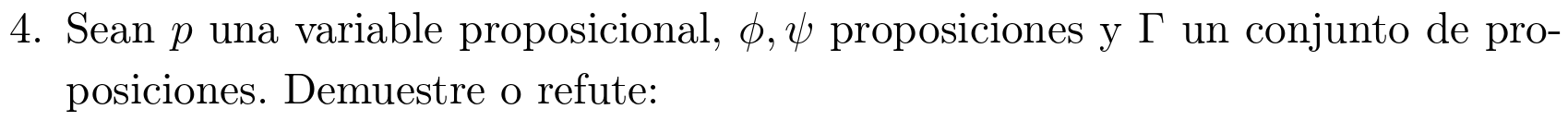
ɸ = ((¬(q → p)) Ʌ (¬((q → p) → p)))

Sección 3.2: 1,4(a),5, 8b



v(ɸ[p := ψ]) = w(ɸ)

(ɸ ≡ ɸ) = (p ≡ p)[p := ɸ]





1. Γ ⊨ ɸ Suposición
2. v satisface a Γ Definición paso 1
3. v(ɸ) = T Definición paso 1
4. v(ɸ[p := ψ]) = w(ɸ) = T Lema 3.6
5. v(ɸ[p := ψ]) = T Transitividad = pasos 3 y 4
6. Salen 2 casos
7. 6.a v satisface a Γ 6.b v no satisface a Γ
8. Γ ⊨ ɸ[p := ψ] Definición pasos 5 y 6a
9. Γ ⊭ ɸ[p := ψ] Definición pasos 5 y 6b
10. Γ ⊭ ɸ[p := ψ] Se toma el paso 9

Así existe un caso donde Γ ⊨ ɸ entonces Γ ⊨ ɸ[p := ψ], no se cumple

Texto, Carta

Descripción generada automáticamentea) ɸ es satisfacible

1. w(ɸ)= T Definición

2. v{ɸ(p:=ψ)} = w(ɸ) Lema 3.6

3. v{ɸ(p := ψ)} = T Transitividad = pasos 2 y 3

4. Así ɸ(p := ψ)} es satisfacible Definición

b) ɸ es insatisfacible

1. w(ɸ)= F Definición

2. v{ɸ(p:=ψ)} = w(ɸ) Lema 3.6

3. v{ɸ(p := ψ)} = F Transitividad = pasos 2 y 3

4. sí ɸ(p := ψ)} es insatisfacible Definición

c) ɸ(p := ψ)} es satisfacible

1. v{ɸ(p:=ψ)} = T Definición

2. v{ɸ(p:=ψ)} = w(ɸ) Lema 3.6

3. w(ɸ)= T Transitividad = pasos 2 y 3

4. Así ɸ es satisfacible Definición

c) ɸ(p := ψ)} es insatisfacible

1. v{ɸ(p:=ψ)} = F Definición

2. v{ɸ(p:=ψ)} = w(ɸ) Lema 3.6

3. w(ɸ)= F Transitividad = pasos 2 y 3

4. Así ɸ es insatisfacible Definición

Texto

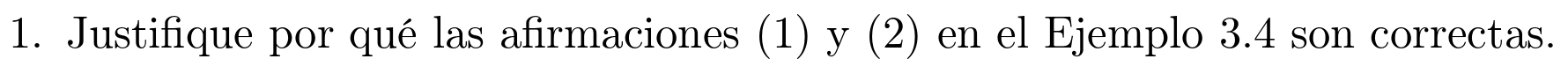
Descripción generada automáticamente

Texto

Descripción generada automáticamente

En algunos casos la sustitución puede fallar, porque se podría llegar a reemplazar 2 veces la misma variable proposicional 2 veces, dando lugar a proposiciones como (p → p), esto cambiando totalmente el (p → q) al que se quería llegar inicialmente

Sección 3.3: 1,8,9





Es cierto por definición de consecuencia tautológica, ya que se toma v(¬ɸ) = T y por meta teorema 2.23 caso ∨, v((¬ɸ) ∨ ψ) = T

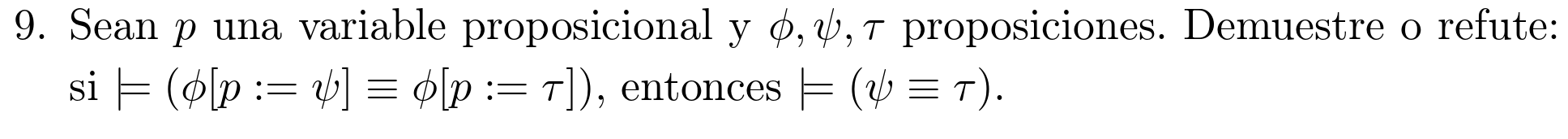


Cuando la valuación de v((¬ɸ) ∨ ψ) = T, se sabe que uno de los casos es que v(ɸ) = F, así al utilizar esta misma valuación en (ɸ → ψ), v(ɸ → ψ) = T, por meta teorema 2.23 caso ≡. Ahora el otro caso, donde v((¬ɸ) ∨ ψ) = F, solo existe un caso y es que v(ɸ) = T y v(ψ) = F, con estas valuaciones ambas partes de (((¬ɸ) ∨ ψ) ≡ (ɸ → ψ)) son falsas, entonces v(((¬ɸ) ∨ ψ) ≡ (ɸ → ψ)) siempre es T

Diagrama

Descripción generada automáticamente con confianza media

Esto es cierto por el teorema 3.10, ya que tan solo es aplicar el teorema n veces



1. ⊨ ( ψ [p ∶= ψ] ≡ ψ [p ∶= τ ]) Suposición
2. Sea ψ = q, τ = false y ψ = true.
3. ⊨ ( q [p ∶= ψ] ≡ q [p ∶= τ ]) ST paso 2
4. ⊨ (q ≡ q) ST paso 3
5. ⊨ (True ≡ False) ST del paso 2
6. v(True ≡ False) = F meta teorema 2.23, caso ≡, paso 5
7. ⊭ ( True ≡ False) Definición paso 6
8. ⊭ (ψ ≡ τ ) ST paso 2

Así si ⊨ (ɸ[p ∶= ψ] ≡ ɸ[p := τ]) entonces hay al menos un caso donde ⊭ (ψ ≡ τ)