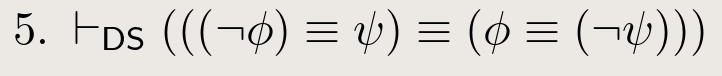
Sección 4.3: 5,6,8,9,10





1. ((ɸ ≢ ψ) ≡ ((¬ɸ) ≡ ψ)) Axioma 10
2. ((¬ɸ) ≡ ψ) ≡ (ψ ≡ (¬ɸ))) Axioma 2, ɸ por (¬ɸ)
3. ((¬ɸ) ≡ ψ) ≡ (ψ ≡ (ɸ ≡ false))) Lema, Axioma 9, Leibniz

ψ por (¬ɸ)

τ por (ɸ ≡ false)

ɸ por [((¬ɸ) ≡ ψ)≡(ψ≡p)]

1. ((¬ɸ) ≡ ψ) ≡ (ψ ≡ (false ≡ ɸ))) Lema, Axioma 2, Leibniz

ψ por (ɸ ≡ false)

τ por (false ≡ ɸ)

ɸ por [((¬ɸ) ≡ ψ)≡(ψ≡p)]

1. ((¬ɸ) ≡ ψ) ≡ ((ψ ≡ false) ≡ ɸ))) Lema, Axioma 1, Leibniz

ψ por (ψ ≡ (false ≡ ɸ)

τ por (ψ ≡ false) ≡ ɸ)

ɸ por [((¬ɸ) ≡ ψ) ≡ p]

1. ((¬ɸ) ≡ ψ) ≡ ((¬ψ) ≡ ɸ))) Lema, Axioma 9, Leibniz

ψ por (ψ ≡ false)

τ por (¬ψ)

ɸ por [((¬ɸ) ≡ ψ) ≡ (p ≡ ɸ))]

1. ((¬ɸ) ≡ ψ) ≡ (ɸ ≡ (¬ψ))) Lema, Axioma 2, Leibniz

ψ por (¬ψ) ≡ ɸ

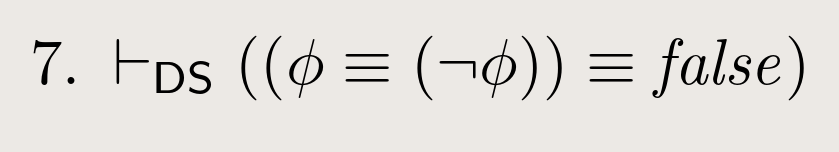
τ por ɸ ≡ (¬ψ)

ɸ por [((¬ɸ) ≡ ψ) ≡ p]

Así 4.15.5 ⊦DS

Texto

Descripción generada automáticamente



1. ((¬ɸ) ≡ (ɸ ≡ false)) Axioma 9
2. (((¬ɸ) ≡ ɸ) ≡ false)) Lema, Regla asociatividad, Leibniz

ψ por (ɸ ≡ false)

τ por (false ≡ ɸ)

ɸ por [(¬ɸ) ≡ p]

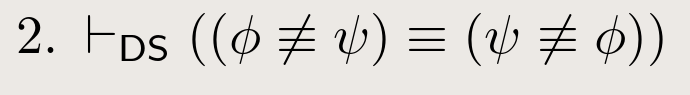
1. ((ɸ ≡ (¬ɸ)) ≡ false)) Lema, Regla conmutatividad, Leibniz

ψ por ((¬ɸ) ≡ ɸ)

τ por ((ɸ ≡ (¬ɸ))

ɸ por [p ≡ false]





1. ((ɸ ≢ ψ) ≡ ((¬ɸ) ≡ ψ)) Axioma 10
2. ((ɸ ≢ ψ) ≡ ((ɸ ≡ false) ≡ ψ)) Lema, Axioma 9, Leibniz

ψ por (¬ɸ)

τ por (ɸ ≡ false)

ɸ por [((ɸ ≢ ψ) ≡ (p ≡ ψ))]

1. ((ɸ ≢ ψ) ≡ ((ɸ ≡ (false ≡ ψ))) Lema, Regla asociatividad, Leibniz

ψ por ((ɸ ≡ false) ≡ ψ))

τ por ((ɸ ≡ (false ≡ ψ))

ɸ por [((ɸ ≢ ψ) ≡ p))]

1. ((ɸ ≢ ψ) ≡ ((ɸ ≡ (¬ψ))) Lema, Axioma 9, Leibniz

ψ por (¬ɸ)

τ por (ɸ ≡ false)

ɸ por [((ɸ ≢ ψ) ≡ (p ≡ ψ))]

1. ((ɸ ≢ ψ) ≡ (((¬ψ) ≡ ɸ)) Lema, Regla conmutatividad, Leibniz

ψ por (ɸ ≡ (¬ψ))

τ por ((¬ψ) ≡ ɸ)

ɸ por [((ɸ ≢ ψ) ≡ p)]

1. ((ɸ ≢ ψ) ≡ (ψ ≢ ɸ)) Lema, Axioma 10, Leibniz

ψ por ((¬ψ) ≡ ɸ)

τ por (ψ ≢ ɸ)

ɸ por [((ɸ ≢ ψ) ≡ p))]

Logotipo

Descripción generada automáticamente con confianza media

Un dibujo de una persona

Descripción generada automáticamente con confianza baja

1. ((ɸ ≢ false) ≡ (false ≢ ɸ)) Teorema 4.16.2, ψ por false
2. ((ɸ ≢ false) ≡ ((¬false) ≡ ɸ)) Axioma 10
3. ((ɸ ≢ false) ≡ (true ≡ ɸ)) Lema: Teorema 4.15.2, Leibniz

ψ por (¬false)

τ por true

ɸ por [((ɸ ≢ false) ≡ (true ≡ p))]

1. ((ɸ ≢ false) ≡ (ɸ ≡ true)) Lema: Regla conmutatividad, Leibniz

ψ por true ≡ ɸ

τ por ɸ ≡ true

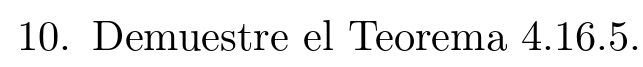
ɸ por [((ɸ ≢ false) ≡ p=]

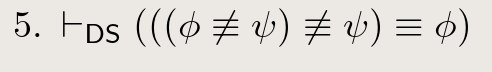
1. ((ɸ ≢ false) ≡ ɸ) Lema: Regla identidad, Leibniz

ψ por (ɸ ≡ true)

τ por ɸ

ɸ por [((ɸ ≢ false) ≡ p)]





1. (((ɸ ≢ ψ) ≢ ψ) ≡ ((¬(ɸ ≢ ψ)) ≡ ψ) Axioma 10, ɸ por (ɸ ≢ ψ)
2. (((ɸ ≢ ψ) ≢ ψ) ≡ ((¬((¬ɸ) ≡ ψ)) ≡ ψ) Lema: Axioma 10, Leibniz

ψ por (ɸ ≢ ψ)

τ por ((¬ɸ) ≡ ψ))

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ (p ≡ ψ))]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ ((¬((ɸ ≡ false) ≡ ψ)) ≡ ψ) Lema: Axioma 9, Leibniz

ψ por (¬ɸ)

τ por (ɸ ≡ false)

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ ((¬((p ≡ ψ)) ≡ ψ)]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ ((¬(ɸ ≡ (false ≡ ψ))) ≡ ψ) Lema: Regla asociatividad, Leibniz ψ por ((ɸ ≡ false) ≡ ψ))

τ por (ɸ ≡ (false ≡ ψ)

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ (((¬p) ≡ ψ)))]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ ((¬(ɸ ≡ (¬ψ)) ≡ ψ) Lema: Axioma 9, Leibniz

ψ por (false ≡ ψ)

τ por (¬ψ)

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ ((¬(ɸ ≡(p)) ≡ ψ]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ (((ɸ ≡ (¬ψ)) ≡ false) ≡ ψ) Lema: Axioma 9, Leibniz

ψ por (¬(ɸ ≡ (¬ψ))

τ por (ɸ ≡ (¬ψ)) ≡ false)

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ (((p) ≡ ψ)]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ (((ɸ ≡ (¬ψ))≡(false ≡ ψ)) Lema: Regla asociatividad, Leibniz

ψ por ((ɸ ≡ (¬ψ)) ≡ false) ≡ ψ)

τ por ((ɸ ≡ (¬ψ))≡(false ≡ ψ))

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ p)]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ (((ɸ≡(¬ψ))≡(¬ψ)) Lema: Axioma 9, conmutatividad, Leibniz

ψ por (false ≡ ψ)

τ por (¬ψ)

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ (((ɸ≡(¬ψ))≡ p)]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ ((ɸ ≡ ((¬ψ) ≡ (¬ψ))) Lema: Regla asociatividad, Leibniz

ψ por (((ɸ≡(¬ψ))≡(¬ψ))

τ por ((ɸ ≡ ((¬ψ) ≡ (¬ψ)))

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ p)]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ (ɸ ≡ true)) Lema: Teorema 4.6.2, Leibniz

ψ por ((¬ψ) ≡ (¬ψ))

τ por true

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ (ɸ ≡ p)]

1. (((ɸ ≢ ψ) ≢ ψ) ≡ ɸ) Lema: Regla identidad, Leibniz

ψ por (ɸ ≡ true)

τ por ɸ

ɸ por [(((ɸ ≢ ψ) ≢ ψ) ≡ p)]

Sección 4.4: 1,2,3,7,10,11,12,13



Texto

Descripción generada automáticamente con confianza baja

1. (((ɸ ∨ (¬true)) ≡ ((¬true) ∨ ɸ)) Axioma 5, ψ por (¬true)
2. (((ɸ ∨ (true ≡ false)) ≡ ((¬true) ∨ ɸ)) Lema: Axioma 9, Leibniz

ψ por (¬true)

τ por (true ≡ false)

ɸ por [(((ɸ ∨ p ≡ ((¬true) ∨ ɸ))]

1. (((ɸ ∨ (true ≡ false)) ≡ (false ∨ ɸ)) Lema: Teorema 4.15.1, Leibniz

ψ por (¬true)

τ por (false)

ɸ por [(((ɸ ∨ (¬true) ≡ (p ∨ ɸ))]

1. (((ɸ ∨ (true ≡ false)) ≡ (ɸ ∨ false)) Lema: Regla conmutatividad, Leibniz

ψ por false ∨ ɸ

τ por ɸ ∨ false

ɸ por [(((ɸ ∨ (true ≡ false)) ≡ p]

1. (((ɸ ∨ (true ≡ false)) ≡ ɸ) Lema: Axioma 6, Leibniz

ψ por (ɸ ∨ false)

τ por ɸ

ɸ por [(((ɸ ∨ (true ≡ false)) ≡ p]

1. (((ɸ ∨ true) ≡ (ɸ ∨ false)) ≡ ɸ) Lema: Axioma 8, Leibniz

ψ por ((ɸ ∨ (true ≡ false))

τ por ((ɸ ∨ true) ≡ (ɸ ∨ false)

ɸ por [p ≡ ɸ]

1. (((ɸ ∨ true) ≡ ɸ) ≡ ɸ) Lema: Axioma 6, Leibniz

ψ por (ɸ ∨ false)

τ por (ɸ)

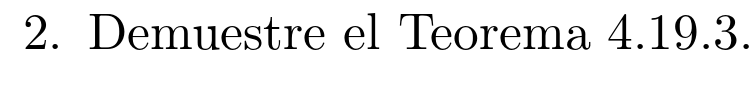
ɸ por [(((ɸ ∨ true) ≡ p) ≡ ɸ)]

1. (((ɸ ∨ true) ≡ (ɸ ≡ ɸ)) Regla asociatividad, ψ, τ por ɸ
2. (((ɸ ∨ true) ≡ true) Lema: 4.6.2, Leibniz

ψ por (ɸ ≡ ɸ)

τ por true

ɸ por [(((ɸ ∨ true) ≡ p]





1. (((ɸ ∨ (¬true)) ≡ ((¬true) ∨ ɸ)) Axioma 5, ψ por (¬true)
2. (((ɸ ∨ (true ≡ false)) ≡ ((¬true) ∨ ɸ)) Lema: Axioma 9, Leibniz

ψ por (¬true)

τ por (true ≡ false)

ɸ por [(((ɸ ∨ p ≡ ((¬true) ∨ ɸ))]

1. (((ɸ ∨ (true ≡ false)) ≡ (false ∨ ɸ)) Lema: Teorema 4.15.1, Leibniz

ψ por (¬true)

τ por (false)

ɸ por [(((ɸ ∨ (¬true) ≡ (p ∨ ɸ))]

1. (((ɸ ∨ (true ≡ false)) ≡ (ɸ ∨ false)) Lema: Regla conmutatividad, Leibniz

ψ por false ∨ ɸ

τ por ɸ ∨ false

ɸ por [(((ɸ ∨ (true ≡ false)) ≡ p]

1. (((ɸ ∨ (true ≡ false)) ≡ ɸ) Lema: Axioma 6, Leibniz

ψ por (ɸ ∨ false)

τ por ɸ

ɸ por [(((ɸ ∨ (true ≡ false)) ≡ p]

1. (((ɸ ∨ true) ≡ (ɸ ∨ false)) ≡ ɸ) Lema: Axioma 8, Leibniz

ψ por ((ɸ ∨ (true ≡ false))

τ por ((ɸ ∨ true) ≡ (ɸ ∨ false)

ɸ por [p ≡ ɸ]

1. (((ɸ ∨ true) ≡ ɸ) ≡ ɸ) Lema: Axioma 6, Leibniz

ψ por (ɸ ∨ false)

τ por (ɸ)

ɸ por [(((ɸ ∨ true) ≡ p) ≡ ɸ)]

1. (((ɸ ∨ true) ≡ (ɸ ≡ ɸ)) Regla asociatividad, ψ, τ por ɸ
2. (((ɸ ∨ true) ≡ true) Lema: 4.6.2, Leibniz

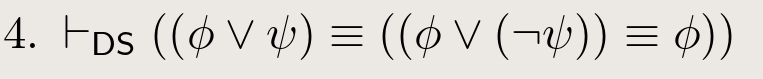
ψ por (ɸ ≡ ɸ)

τ por true

ɸ por [(((ɸ ∨ true) ≡ p]

1. (ɸ ∨ true) Regla identidad, ɸ por (ɸ ∨ true)





1. ((ɸ ∨ (¬ψ)) ≡ ((¬ψ) ∨ ɸ)) Axioma 5, ψ por (¬ψ)
2. ((ɸ ∨ (ψ ≡ false)) ≡ ((¬ψ) ∨ ɸ)) Lema: Axioma 9, Leibniz

ψ por (¬ψ)

τ por ((ψ ≡ false))

ɸ por [((ɸ ∨ p)≡ ((¬ψ) ∨ ɸ))]

1. (((ɸ ∨ ψ) ≡ (ɸ ∨ false)) ≡ ((¬ψ) ∨ ɸ)) Lema: Axioma 8, Leibniz

ψ por (ɸ ∨ (ψ ≡ false)

τ por ((ɸ ∨ ψ) ≡ (ɸ ∨ false))

ɸ por [p ≡ ((¬ψ) ∨ ɸ))]

1. (((ɸ ∨ ψ) ≡ (ɸ ∨ false)) ≡ (ɸ ∨ (¬ψ))) Lema: Regla conmutatividad, Leibniz

ψ por ((¬ψ) ∨ ɸ)

τ por (ɸ ∨ (¬ψ))

ɸ por [(((ɸ ∨ ψ) ≡ (ɸ ∨ false)) ≡ p]

1. (((ɸ ∨ ψ) ≡ ɸ) ≡ (ɸ ∨ (¬ψ))) Lema: Axioma 6, Leibniz

ψ por (ɸ ∨ false)

τ por (ɸ)

ɸ por (((ɸ ∨ ψ) ≡ p ) ≡ (ɸ ∨ (¬ψ))

1. (((ɸ ∨ ψ) ≡ (ɸ ≡ (ɸ ∨ (¬ψ))) Regla asociatividad, ɸ por (ɸ ∨ ψ)

ψ por ɸ

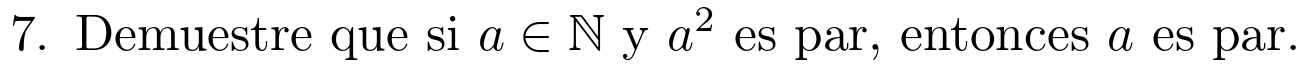
τ por (ɸ ∨ (¬ψ))

1. ((ɸ ∨ ψ) ≡ ((ɸ ∨ (¬ψ)) ≡ ɸ)) Lema: Regla conmutatividad, Leibniz

ψ por (ɸ ≡ (ɸ ∨ (¬ψ))

τ por (ɸ ∨ (¬ψ)) ≡ ɸ)

ɸ por [((ɸ ∨ ψ) ≡ p]



Demostración por contradicción

Suponemos el recíproco de la conclusión

a es impar, entonces a se puede escribir de la forma: a = 2k + 1

Así a2 = (2k + 1)2

a2 = (4k2 + 4k +1)

a2 = 4k(4k + 2)

a2 = 4k(m) m = 4k + 2

a2 = 2 ∙ 2km

a2 = 2(2km)

a2 = 2n n = 2km

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. (ɸ ∨ ψ) Suposición
2. (¬ɸ) Suposición
3. ((¬ɸ) ≡ (ɸ ≡ false)) Axioma 9
4. ((¬ɸ) ≡ (false ≡ ɸ)) Lema: Regla conmutatividad, Leibniz

ψ por (ɸ ≡ false)

τ por (false ≡ ɸ)

ɸ por [(¬ɸ) ≡ p]

1. ((¬ɸ) ≡ false) ≡ ɸ)) Regla conmutatividad
2. (((¬ɸ) ≡ false) ∨ ψ) ≡ (ɸ ∨ ψ))) Leibniz en 1, ψ por (¬ɸ) ≡ false

τ por ɸ

ɸ por [p ∨ ψ]

1. (((¬ɸ) ≡ false) ∨ ψ) Ecuanimidad\* 1 y 6
2. ((¬ɸ) ≡ true) Regla identidad en 2
3. (((¬ɸ) ≡ false) ∨ ψ) ≡ (true ≡ false) ∨ ψ)) Leibniz en 8,

ψ por (¬ɸ)

τ por true

ɸ por [((p ≡ false) ∨ ψ)]

1. ((true ≡ false) ∨ ψ) Ecuanimidad 7 y 9
2. ((true ≡ false) ≡ (false ≡ true)) Axioma 2, ɸ por true, ψ por false
3. (((true ≡ false) ∨ ψ) ≡ ((false ≡ true) ∨ ψ)) Leibniz en 11

ψ por (true ≡ false)

τ por (false ≡ true)

ɸ por [p ∨ ψ]

1. ((false ≡ true) ∨ ψ) Ecuanimidad 10 y 12
2. ((false ≡ true) ≡ false) Axioma 3, ɸ por false
3. (((false ≡ true) ∨ ψ) ≡ (false ∨ ψ)) Leibniz en 11, ψ por (false ≡ true)

τ por false

ɸ por [p ∨ ψ]

1. (false ∨ ψ) Ecuanimidad 13 y 15
2. (ψ ∨ false) Regla conmutatividad en 16
3. ψ Axioma 6

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. (ɸ ∨ ψ) Suposición
2. ((¬ɸ) ∨ τ) Suposición
3. (((¬(¬ɸ)) ∨ τ) ≡ ɸ) Teorema 4.19.4, en 2 ɸ por (¬ɸ)
4. (ɸ ≡ ((¬(¬ɸ)) ∨ τ)) Regla conmutatividad en 3
5. ((¬(¬ɸ)) ≡ ɸ)) Teorema 4.15.6
6. ((¬(¬ɸ)) ∨ τ) ≡ (ɸ ∨ τ))) Leibniz en 4, ψ por (¬(¬ɸ))

τ por ψ

ɸ por [p ∨ τ]

1. (ɸ ≡ (ɸ ∨ τ)) Transitividad 4 y 6
2. ((ɸ ∨ ψ) ≡ ((ɸ ∨ τ) ∨ ψ)) Leibniz en 7, ψ por ɸ

τ por (ɸ ∨ τ)

ɸ por [p ∨ ψ]

1. ((ɸ ∨ τ) ∨ ψ)) Ecuanimidad 1 y 8
2. ((ɸ ∨ (τ ∨ ψ)) Axioma 4 en 9, ψ por τ

τ por ψ

1. ((ψ ∨ τ) ≡ (τ ∨ ψ)) Axioma 5 ɸ por ψ, ψ por τ
2. (ɸ ∨ (ψ ∨ τ) ≡ ɸ ∨ (τ ∨ ψ)) Leibniz en 11, ψ por (ψ ∨ τ)

τ por (τ ∨ ψ)

ɸ por [ɸ ∨ p]

1. (ɸ ∨ (ψ ∨ τ)) Ecuanimidad\* 10 y 12
2. ((ɸ ∨ ψ) ∨ τ)) Axioma 5 en 13
3. ((ɸ ∨ ψ) ∨ (¬τ))) Teorema 4.19.4 en 14, ɸ por (ɸ ∨ ψ)

ψ por τ

1. ((¬τ) ≡ (τ ≡ false)) Axioma 9, ɸ por τ
2. (((ɸ ∨ ψ) ∨ (¬τ)) ≡ ((ɸ ∨ ψ) ∨ (τ ≡ false)) Leibniz, ψ por (¬τ)

τ por (τ ≡ false)

ɸ por [((ɸ ∨ ψ) ∨ p)]

1. (((ɸ ∨ ψ) ∨ (τ ≡ false)) Ecuanimidad 15 y 17
2. (((ɸ ∨ ψ) ∨ τ) ≡ ((ɸ ∨ ψ) ∨ false))) Axioma 8, ɸ por (ɸ ∨ ψ)

ψ por (τ)

τ por false

1. (((ɸ ∨ ψ) ∨ false) ≡ (ɸ ∨ ψ)) Axioma 6, ɸ por (ɸ ∨ ψ)
2. (((ɸ ∨ ψ) ∨ τ) ≡ ((ɸ ∨ ψ) ∨ false)) ≡ (((ɸ ∨ ψ) ∨ τ) ≡ (ɸ ∨ ψ))) Leibniz en 20

ψ por ((ɸ ∨ ψ) ∨ false)

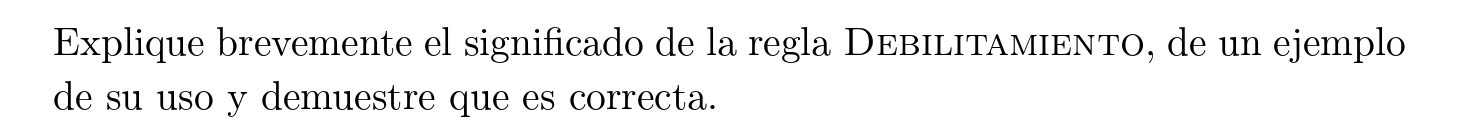
τ por (ɸ ∨ ψ)

ɸ por [((ɸ ∨ ψ) ∨ τ) ≡ p]

1. (((ɸ ∨ ψ) ∨ τ) ≡ (ɸ ∨ ψ)) Ecuanimidad 19 y 21
2. (ɸ ∨ ψ) Ecuanimidad 13 y 22

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



1. ɸ Suposición
2. (ɸ ≡ true) Regla identidad
3. ((ɸ ∨ ψ) ∨ (true ≡ ψ)) Leibniz, ψ por ɸ

τ por true

ɸ por [p ∨ ψ]

1. ((true ≡ ψ) ≡ (ψ ≡ true)) Axioma 2, ɸ por true
2. ((ψ ≡ true) ≡ ψ) Regla identidad, ɸ por ψ
3. ((true ≡ ψ) ≡ ψ) Transitividad 4 y 5
4. (((ɸ ∨ ψ) ∨ (true ≡ ψ)) ≡ ((ɸ ∨ ψ) ∨ ψ)) Leibniz en 6

ψ por (true ≡ ψ)

τ por ψ

ɸ por [((ɸ ∨ ψ) ∨ p)]

1. ((ɸ ∨ ψ) ∨ ψ) Ecuanimidad 3 y 7
2. ((ɸ ∨ (ψ ∨ ψ)) Axioma 4, τ por ψ
3. ((ψ ∨ ψ) ≡ ψ) Axioma 7, ɸ por ψ
4. ((ɸ ∨ (ψ ∨ ψ)) ≡ (ɸ ∨ ψ)) Leibniz en 10, ψ por (ψ ∨ ψ)

τ por ψ

ɸ por [ɸ ∨ p]

1. (ɸ ∨ ψ) Ecuanimidad 9 y 11

Imagen de la pantalla de un celular con texto

Descripción generada automáticamente con confianza media

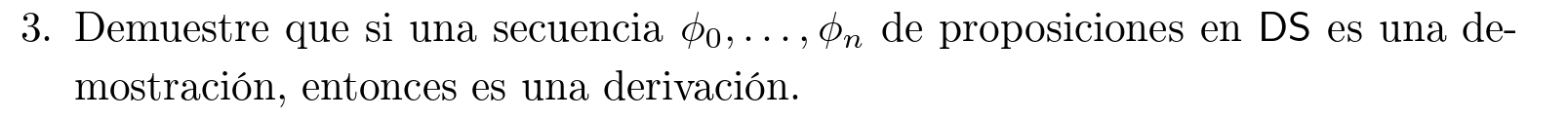
Se tiene el caso donde {ɸ ⟼ false, ψ ⟼ true}

Así

1. (false ∨ true) Suposición
2. ((false ∨ true) ≡ true) Teorema 4.19.2
3. true Ecuanimidad 1 y 2
4. ((¬false) ≡ true) Teorema 4.15.2
5. (false ≡ (¬false)) ≡ false Teorema 4.15.7, ɸ por false
6. false

Entonces, esta regla es incorrecta, ya que true no equivale a false, entonces la conclusión es falsa

Sección 4.5: 3,4,5,6

ɸ0

≡ ”¿Por qué (ɸ0 ≡ ɸ1)?”

ɸ1

≡ ”¿Por qué (ɸ1 ≡ ɸn-1)?”

.

.

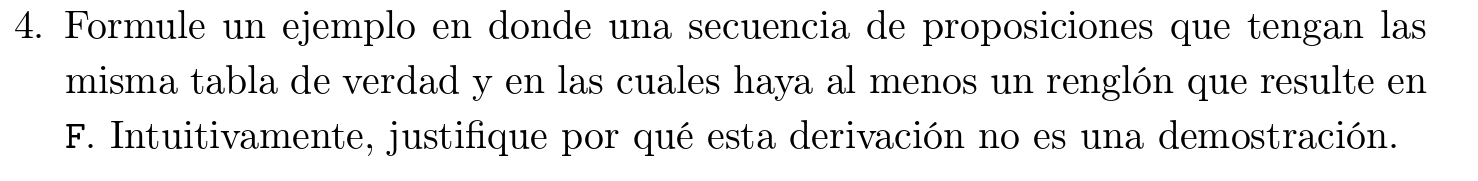
.

ɸn-1

≡ ”¿Por qué (ɸn-1 ≡ ɸn)?”

ɸn

Es una derivación, ya que tiene la misma estructura secuencial además de que ambas tienen como objetivo llegar a un teorema, por lo tanto, se puede usar una forma para escribir la otra



(ɸ ≡ true)

La tabla de verdad de esta proposición es:

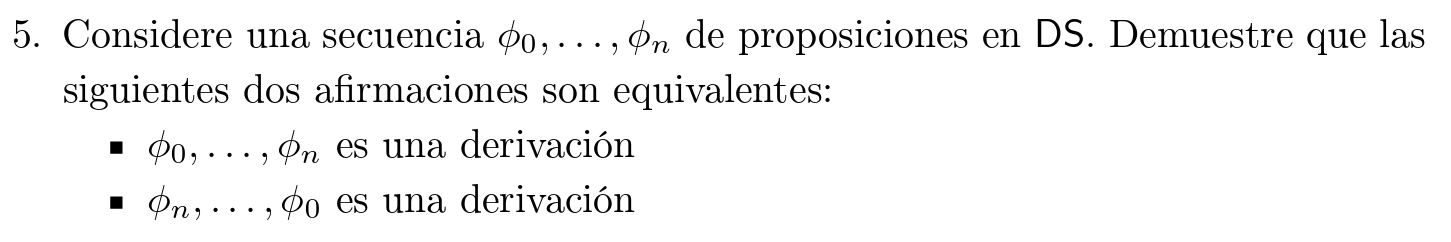
|  |  |  |
| --- | --- | --- |
| ɸ | true | (ɸ ≡ true) |
| T | T | T |
| F | T | F |

(ψ ≡ true)

La tabla de verdad de esta proposición es:

|  |  |  |
| --- | --- | --- |
| ψ | true | (ψ ≡ true) |
| T | T | T |
| F | T | F |

Aunque estas 2 proposiciones tengan la misma tabla de verdad, no se puede decir que (ɸ ≡ ψ), ya que hay un caso donde da falso, y es donde ambos tienen como valor de verdad, false



La primera afirmación es verdadera por el teorema 4.21

Por la regla de conmutatividad usada en el ejemplo 4.4, la derivación puede empezar en cualquiera de los 2 sentidos, es decir de derecha a izquierda o de izquierda a derecha, ya que por esta regla, está correcto

Texto

Descripción generada automáticamente

Estas 2 afirmaciones son ciertas, ya que la segunda es una derivación, escrita en una sola línea, ya que los puntos indican que hay un proceso entre ambas equivalencias, por lo tanto una demostración y ese es el objetivo de una derivación