Sección 6.1: 4

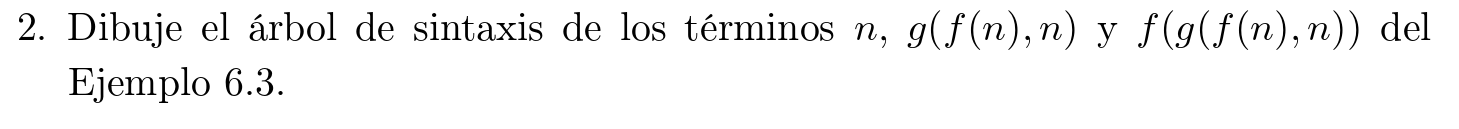
Texto

Descripción generada automáticamente

La lógica aristotélica es un sistema formal de razonamiento y argumentación desarrollado por el filósofo griego Aristóteles en el siglo IV a.C. Esta lógica se basa en la clasificación de los juicios y proposiciones en cuatro tipos (A, E, I y O) y en la utilización de reglas para inferir conclusiones válidas a partir de premisas dadas.

La lógica aristotélica se enfoca en la estructura y la validez de los argumentos y no en su contenido, y se utiliza para analizar y evaluar la coherencia de los argumentos y las proposiciones. Es considerada una de las primeras y más influyentes formas de lógica en la historia de la filosofía occidental, y su legado ha sido fundamental para el desarrollo de la lógica moderna.

Sección 6.2: 2, 3



1. n

2. g

f n

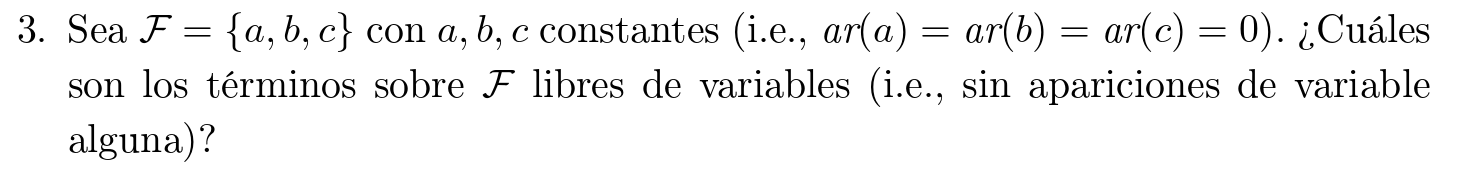
n

3. f

g

f n

n



Sección 6.3: 8, 10, 11, 12, 14, 16, 19

Texto, Carta

Descripción generada automáticamente

1. (∀x ∣: A(m, y))
2. (∃x ∣: (P(x) Ʌ A(x, m)))
3. A(m, m)
4. (¬∀x ∣:(C(x) Ʌ B(y, x))
5. (¬∃x ∣:(C(x) Ʌ B(y, x)))
6. (¬∃x |: (∀y |C(x) ∧ E(y) : B(y, x)))

Texto, Carta

Descripción generada automáticamente

1. ∀x ∣:(M(x, y))
2. ∀x ∣:(M(x, y) Ʌ P(x, y))
3. ∀x ∣(∃x∣:M(x, y) : ∃x∣:P(x, y))
4. ∃x ∣:(P(j, x) : (∃x∣:P(x, z) ∨ M(x, z))
5. ∃j ∣:(P(x, j) Ʌ P(y, ja) Ʌ H(x, y))
6. ∃x ∣:(M(x, y) Ʌ H(x, z) Ʌ M(z, m))
7. ¬∀x ∣(H(x, y) Ʌ P(x, z) Ʌ ¬P(y, z))
8. (∃x ∣: (∃y ∣M(x,y): (¬∃z ∣ M(y,z) ∨ P(y,z)): (∃w ∣:P(x,w)
9. E(j,ja)
10. ∀x ∣(H(m,x) Ʌ E(x,c))

Texto

Descripción generada automáticamente

**Predicados:**  
P(x, y): "x y “y” son iguales"

1. (∃x|:(∃y│: ¬P(x, y) )
2. (∃x│:(∃y|: ¬P(x, y) ∨ P(x, y)))
3. (∃x|:( ∃y │: (∃z │: ¬P(x, y) ∧ ¬P(x, w) ∧ ¬P(w, y)))))
4. (∀x |: (∀y|: (∃w│: P(x, y) ∨ ¬P(x, y) : ¬P(w, x) ∧ ¬P(w, y))))

Texto

Descripción generada automáticamente

1. ∃x ∣: R(x) : (∃y ≠ x ∣: R(y))
2. ∀x ∣: (R(x) : (∃y ∣: ¬R(y)) : (∃z ∣: ¬R(z))

Texto

Descripción generada automáticamente

Predicados

H(x) = “x es humano”

E(x) = “x es egoísta”

1. ∀x∣:(H(x) Ʌ E(x))
2. ¬∀x∣:(H(x) Ʌ E(x))
3. ∃x∣:(H(x) Ʌ E(x))
4. ∃x∣:(H(x) Ʌ ¬E(x))

Texto

Descripción generada automáticamente

Funciones:

u : “Usted”

Predicados

E(x,y) : “x engaña a y”

1. (∃x|:E(u, x)∨¬E(u, x)))
2. (∀x|:E(u, x)∨¬E(u, x)))
3. (¬∀x|:E(u, x)∨¬E(u, x)))
4. (∃x│:¬P(u, x))

Texto, Carta

Descripción generada automáticamente

Para expresar esta proposición en lógica de predicados, podemos utilizar los siguientes conjuntos de símbolos:

Funciones:

tarea(t): Devuelve el tiempo en segundos que tarda en completarse la tarea t.

inicio(t): Devuelve el tiempo más temprano en el cual se puede iniciar la tarea t.

Predicados:

completada(t): Indica si la tarea t ha sido completada.

prer(t): Devuelve la colección de tareas prerrequisito de la tarea t.

todas\_completadas(c): Indica si todas las tareas en la colección c han sido completadas.

Entonces, podemos expresar la proposición de la siguiente manera:

Para cada tarea t, si todas las tareas prerrequisito en la colección prer(t) han sido completadas, entonces el tiempo de inicio de la tarea t es el tiempo más temprano en el cual todas las tareas prerrequisito en la colección prer(t) han sido completadas y se debe tomar en cuenta el tiempo que tarda la tarea t en completarse:

∀t (todas\_completadas(prer(t)) → inicio(t) = min{inicio(s) + tarea(s) | s ∈ prer(t)})

La expresión "min{inicio(s) + tarea(s) | s ∈ prer(t)}" indica que se debe buscar el valor mínimo de inicio(s) + tarea(s) para todas las tareas s en la colección prer(t), y luego se debe asignar este valor como el tiempo de inicio de la tarea t.

Sección 6.4: 3,4,5,6

Texto, Carta

Descripción generada automáticamente

Formulas Variables libres Variables acotadas

S(m, x) x Ninguna

B(m, F(m)) m Ninguna

B(x, y) → ∃z S(z, y) x, y z

∃y B(x, y) → ∃z S(z, y) x y, z

S(x, y) → S(y, f(f(x))) x, y Ninguna

Texto, Carta

Descripción generada automáticamente

Libres:

y

Acotadas:

x

Alcances

∀x P(f(d),h(h(c, x), d), y)

1. .

Libres:

x

Acotadas:

y, z

Alcances

, 

Libres:

x, z

Acotadas:

y

Alcances

, 

Libres:

x, z

Acotadas:

y

Alcances

, 

Libres:

Ninguna

Acotadas:

x, y, z

Alcances

, 

Libres:

Libres:

Ninguna

Acotadas:

x, y, z

Alcances

, 

Libres:

Ninguna

Acotadas:

x, y, z

Alcances

, 

Texto, Carta

Descripción generada automáticamente

1. ∃x

Ʌ

P ∀y

y z ∨

¬ P

Q y z

y x

Libres:

z

Acotadas:

x, y

“y” es el caso de la variable que está libre y acotada, ya que por la izquierda está libre, pero por la derecha está acotada



P(y, z)



El alcance de ∃x ya es P(y, z)

Texto, Carta

Descripción generada automáticamente

Si ϕ es una fórmula, entonces quant(x, ϕ)es cierta si y solo si   
ϕ contiene una instancia de x cuantificada.

Si ϕ es una fórmula, entonces quant(x, ϕ)es cierta si y solo si alguna de las siguientes condiciones se cumple:

a. ϕ es de la forma ¬ψ y quant(x, ψ) es cierta.

b. ϕ es de la forma ψ ∧ ɸ y quant(x, ψ) y quant(x, ɸ) son ciertas.

c. ϕ es de la forma ψ ∨ ɸ y quant(x, ψ) o quant(x, ɸ) son ciertas.

d. ϕ es de la forma ψ → ɸ y quant(x, ψ) o quant(x, ɸ) son ciertas.

e. ϕ es de la forma ∀ y ψ y “y” no es igual a x y quant(x, ψ) es cierta.

f. ϕ es de la forma ∃ y ψ y “y” no es igual a x y quant(x, ψ) es cierta.