Tarea LCAT #2

Juan Sebastián Velásquez Rodríguez

Sección 1.1

Texto

Descripción generada automáticamente

Texto

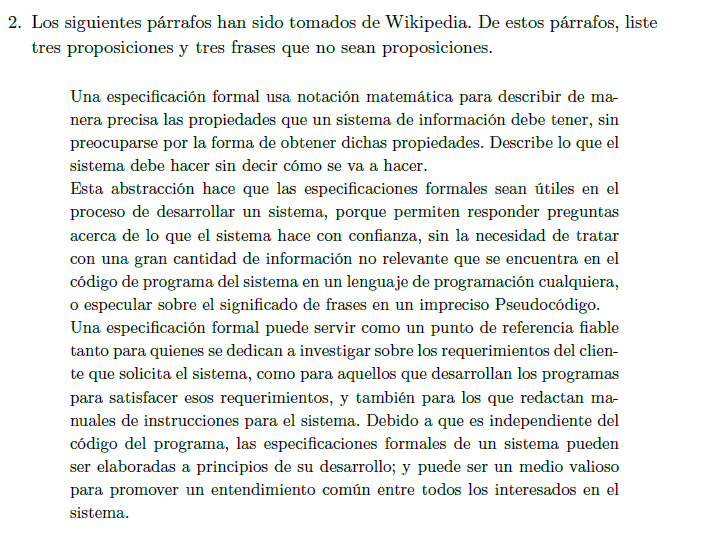
Descripción generada automáticamente

No proposiciones

* Sólo de demostraciones y razonamientos que pueden ser completamente formalizados en todos sus aspectos.
* Las capacidades expresivas de los diferentes lenguajes formales y las propiedades metalógicas de los mismos
* De clasificar su capacidad expresiva

Proposiciones

* La teoría de la demostración y la matemática inversa son dos de los razonamientos más recientes de la lógica matemática abstracta.
* La lógica matemática estudia los sistemas formales en relación con el modo en el que codifican conceptos intuitivos de objetos matemáticos como conjuntos, números, demostraciones y computación.
* La lógica estudia las reglas de deducción formales, las capacidades expresivas de los diferentes lenguajes formales y las propiedades metalógicas de los mismos.



No proposiciones

* Especular sobre el significado de frases en un impreciso Pseudocódigo.
* Como para aquellos que desarrollan los programas para satisfacer esos requerimientos,
* También para los que redactan manuales de instrucciones para el sistema.

Proposiciones

* Una especificación formal usa notación matemática para describir de manera precisa las propiedades que un sistema de información debe tener, sin preocuparse por la forma de obtener dichas propiedades.
* Una especificación formal puede servir como un punto de referencia fiable tanto para quienes se dedican a investigar sobre los requerimientos del cliente que solicita el sistema, como para aquellos que desarrollan los programas para satisfacer esos requerimientos, y también para los que redactan manuales de instrucciones para el sistema.
* Esta abstracción hace que las especificaciones formales sean útiles en el proceso de desarrollar un sistema, porque permiten responder preguntas acerca de lo que el sistema hace con confianza, sin la necesidad de tratar con una gran cantidad de información no relevante que se encuentra en el código de programa del sistema en un lenguaje de programación cualquiera, o especular sobre el significado de frases en un impreciso Pseudocódigo.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. (p -> q), p, q
2. ((¬p) -> q), (¬p), q
3. p, q, r

a y b, tienen la misma estructura

Sección 1.2

1b) (true≡false)

Es una proposición, porque: true y false son proposiciones, a su vez, una equivalencia entre 2 proposiciones, sigue siendo proposición

1d) (p V (p ≡ (¬q)))

Es una proposición, porque, ¬q, es una proposición, cualquier equivalencia entre 2 proposiciones, se vuelve una proposición, al ser esto una proposición, la conjunción entre 2 proposiciones es otra proposición

1f) ((q Ʌ (¬q)) <- (¬(¬(¬(p -> r)))))

P -> r, es una proposición que implica a otra, esto sigue siendo una proposición, la negación de una proposición, siempre será una proposición, sin importar cuantas veces se niegue esta, q Ʌ (¬q), es una disyunción de proposiciones, sigue siendo una proposición y por último la consecuencia de proposiciones es una proposición

2a) (pV)

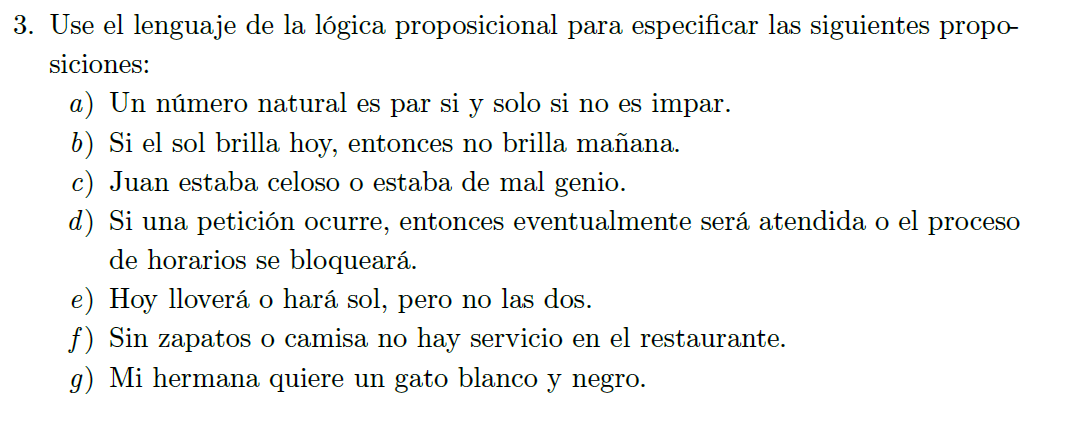
Esto no es una proposición, porque la conjunción, necesita de 2 proposiciones, una a cada lado, y en este caso solo hay una

2c) ¬p

No es una proposición porque no tiene los paréntesis que encierran la proposición, cosa que es fundamental en las negaciones para no dar pie a malinterpretaciones

2e) (p V q) V r

Al igual que el punto anterior, faltan los paréntesis para saber donde termina y empieza la proposición, para no confundirse



a) ((p <- q)≡(¬r))

p= Un número es natural

q= Un número es par

r= Un número es impar

b) (p -> (¬q))

p = El sol brilla hoy

q = El sol no brilla mañana

c) (p V q)

p = Juan está celoso

q = Juan está de mal genio

d) ((p -> q) V r)

p = Una petición ocurre

q = La petición eventualmente será atendida

r = El proceso de horarios se bloqueará

e) ((p V q) ≠ (p Ʌ q))

p = Hoy lloverá

q = Hará sol

f) (((¬p) V (¬q)) -> (¬r))

p = Tiene zapatos

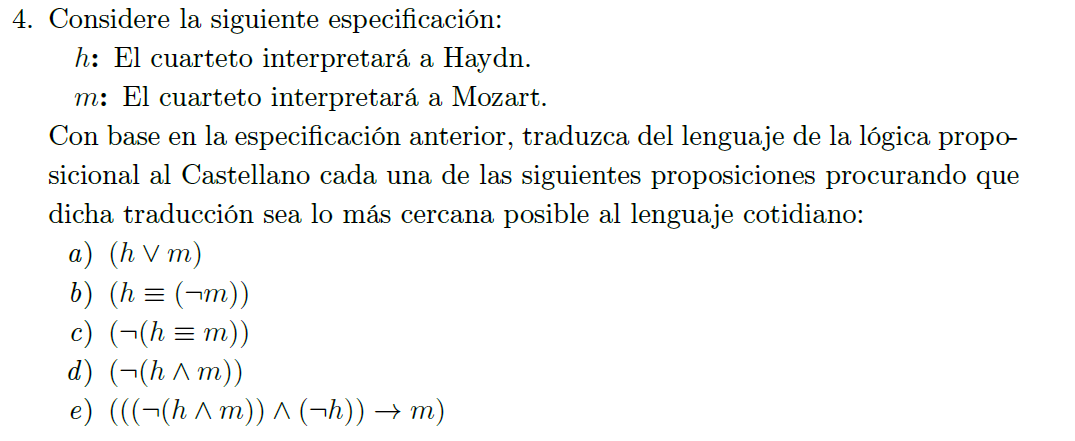
q = Tiene camisa

r = Hay servicio en el restaurante

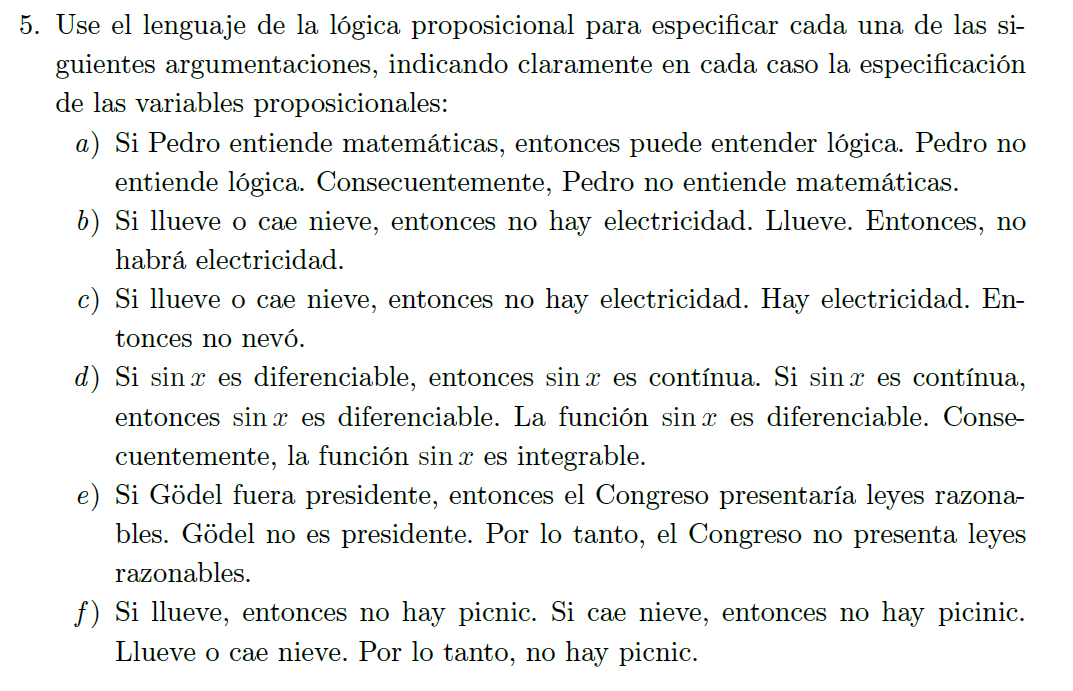
e) ((p Ʌ q) ≠ (p V q))

p = mi hermana quiere un gato blanco

q = mi hermana quiere un gato negro



1. El cuarteto interpretará a Haydn o Mozart
2. El cuarteto interpretará a Haydn pero no a Mozart
3. Si el cuarteto no interpreta a Mozart, tampoco interpretará a Haydn
4. El cuarteto no interpretará ni a Haydn ni a Mozart
5. Si el cuarteto no interpreta a Haydn ni a Mozart y no interpreta a Haydn , entonces interpretará a Haydn



1. (p -> q), (¬q), (¬p)

p = Pedro entiende matemáticas

q = Pedro puede entender lógica

1. ((p V q) -> (¬r)), (p), (¬r)

p = Está lloviendo

q = está nevando

r = Hay electricidad

1. ((p V q) -> (¬r)), (r), (¬q)

p = Está lloviendo

q = está nevando

r = Hay electricidad

1. (p -> q), (q -> p), (p),(r)

p = sinx es diferenciable

q = sinx es continua

r = sinx es integrable

1. (p -> q), (¬p), (¬q)

p = Gödel es presidente

q = El congreso presenta leyes razonables

1. (p -> (¬q)), (r -> (¬q)), (p V r),(¬q)

p = Está lloviendo

q = Hay picnic

r = Está cayendo nieve