Luis Arturo Izaguirre Viera
lsizaguirre@gmail.com
02MIAR Matemáticas para la IA

NOTA: Este documento forma parte de la entrega de los ejercicios evaluables de la asignatura de Matemáticas para la Inteligencia Artificial. En este documento solo se resuelven algunos de los ejercicios planteados y se entrega junto a un notebook (.ipynb) anexo donde se desarrollan el resto de ejercicios.

1. Primer ejercicio

Demuéstrese o refútese razonadamente la equivalencia entre los pares de enunciados siguientes:

1.1. Primer par de enunciados

- $\neg\exists x : \neg(\neg p(x) \vee \neg q(x))$, al que llamaremos primer enunciado
- $[\forall x, \neg p(x)] \vee [\forall x, \neg q(x)]$, al que llamaremos segundo enunciado

Solución:

a. Para demostrar o refutar su equivalencia partiremos aplicando al primer enunciado la siguiente ley de Morgan:
$$\forall x, p(x) \equiv \neg[\exists x : \neg p(x)]$$

b. De esta forma aplicando la ley descrita en el paso anterior sobre el primer enunciado $\neg\exists x : \neg(\neg p(x) \vee \neg q(x))$, podríamos también escribirlo así:
$$\forall x : \neg p(x) \vee \neg q(x)$$

c. Además sabemos que existe una propiedad de la equivalencia que nos dice:
$$[\forall x, p(x)] \vee [\forall x, q(x)] \models \forall x : [p(x) \vee q(x)]$$

d. Tomando esto último en consideración podríamos aplicarlo al primer enunciado y quedaría demostrado que:
$$[\forall x, \neg p(x)] \vee [\forall x, \neg q(x)] \models \forall x : [\neg p(x) \vee \neg q(x)]$$

e. Por esto último podemos concluir que el primer enunciado implica al segundo, pero no en sentido contrario, por lo tanto, **NO son equivalentes**, el primer enunciado implica el segundo pero el segundo no implica el primero.

1.2. Segundo par de enunciados

- $\neg(\forall x, \exists y : [(p(x,y) \wedge q(x,y)) \implies r(x,y)])$, al que llamaremos primer enunciado
- $\exists x : [(\forall y, p(x,y)) \wedge (\forall y, q(x,y)) \wedge (\forall y, \neg r(x,y))]$, al que llamaremos segundo enunciado

Solución:

a. Analizando inicialmente ambos enunciados decido partir por el segundo, intentando resolver primero la parte interna del mismo. A esta parte la llamaremos AUX, de esta forma podríamos reescribir el segundo enunciado como:
$$\exists x : AUX,$$
donde AUX = $[(\forall y, p(x,y)) \wedge (\forall y, q(x,y)) \wedge (\forall y, \neg r(x,y))]$

b. Ahora nos concentraremos en simplificar AUX y para ello aplicaremos la siguiente propiedad:
$$[\forall x, p(x)] \wedge [\forall x, q(x)] \equiv \forall x : [p(x) \wedge q(x)]$$

c. Aplicando lo anterior sobre AUX = $[(\forall y, p(x,y)) \wedge (\forall y, q(x,y)) \wedge (\forall y, \neg r(x,y))]$, AUX nos quedaría de la siguiente forma:
$$\forall y, [p(x,y)) \wedge q(x,y) \wedge \neg r(x,y)]$$

d. Además sabemos que existe una ley de Morgan que nos dice:
$$\forall x, p(x) \equiv \neg[\exists x : \neg p(x)]$$

e. Si aplicamos esta ley sobre AUX, tendríamos que AUX sería igual a:
$$\neg\exists y, \neg[p(x,y)) \wedge q(x,y) \wedge \neg r(x,y)]$$

f. También sabemos que:
$$a \wedge (b \wedge c) \equiv (a \wedge b) \wedge c$$
y que
$$\neg(a \wedge b) \equiv \neg a \vee \neg b$$

g. Si consideramos:
$$a = p(x,y)) \wedge q(x,y),$$
$$b = \neg r(x,y)$$

h. Podríamos perfectamente reescribir AUX como:
$$\neg\exists y, [\neg(p(x,y)) \wedge q(x,y)) \vee \neg(\neg r(x,y))]$$
o lo que es lo mismo:
$$\neg\exists y, [\neg(p(x,y)) \wedge q(x,y)) \vee r(x,y)]$$

i. Para finalizar con la manipulación de AUX, también recordemos que $\neg p \vee q \equiv p \implies q$, y aplicando esto sobre AUX quedaría de la siguiente forma:
$$\neg\exists y, [(p(x,y)) \wedge q(x,y)) \implies r(x,y)]$$

j. Ahora podemos sustituir AUX en su original, es decir, en el segundo enunciado:
$$\exists x : AUX,$$
$$\exists x : \neg\exists y, [(p(x,y)) \wedge q(x,y)) \implies r(x,y)]$$

k. Para finalizar, aplicaremos sobre el cuantificador existencial de x la siguiente ley:
$$\exists x : p(x) \equiv \neg[\forall x, \neg p(x)]$$

l. Quedando el segundo enunciado reescrito de la siguiente forma:
$$\neg(\forall x, \exists y : [(p(x,y) \wedge q(x,y)) \implies r(x,y)])$$

m. lo que es igual al primer enunciado, por lo tanto estos dos enunciados **SON EQUIVALENTES**

2. Segundo ejercicio

Sea $A \in \mathbb{R}^{3x3}$ la matriz cuadrada dada por
$$A = \begin{pmatrix} 3 & -3 & 2 \\ -4 & 4 & -4 \\ -3 & 3 & -2 \end{pmatrix}$$

2.1. Obténgase cuatro matrices $B \in \mathbb{R}^{3x3}$ tales que $B^2 = A$. Sugerencia: diagonalizar A puede ser de utilidad. Para el proceso de diagonalización, puedes hacer uso de `numpy.linalg.eig` en Python.

Este apartado se encuentra resuelto paso a paso en el notebook anexo debido a que se implementa haciendo uso de código Python.

2.2. ¿Crees que las matrices se podrían haber obtenido por tanteo, empleando la fuerza bruta, con un coste computacional similar? ¿Qué nos dice esto, en términos de optimización, acerca de utilizar en determinadas ocasiones estrategias matemáticas en la resolución computacional de problemas?

Con un coste computacional similar, nunca. Hacerlo a través de la fuerza bruta nos obligaría a probar iterando números para cada posición en la matriz, para realizar una búsqueda aleatoria en una matriz $\mathbb{N}x\mathbb{N}$ en un espacio de números enteros entre 0 y 100 deberíamos iterar $100^{\mathbb{N}x\mathbb{N}}$ veces.

Esto nos hace pensar que el uso de estrategias matemáticas en la resolución de problemas computacionales en la mayoría de las oportunidades nos permitirá optimizar los algoritmos que utilicemos bajo reglas y condiciones que acotan el problema y lo hacen más sencillo de solucionar.

3. Tercer ejercicio

Este apartado se encuentra resuelto paso a paso en el notebook anexo debido a que se implementa haciendo uso de código Python.

4. Cuarto ejercicio

Se dispone de tres cartas con las características siguientes:

- La primera carta es roja por ambos lados.
- La segunda carta es verde por ambos lados.
- La tercera carta es roja por un lado y verde por el otro lado.

Se introducen las tres cartas en una bolsa, se barajan y se extrae una carta completamente al azar, poníendola sobre la mesa sin mirar y observando posteriormente el lado que ha quedado visible, el cual ha resultado ser de color verde.

a. **Intuitivamente, ¿cuál crees que es la probabilidad de que al girar la carta el otro lado resulte ser también verde?**

   Lo habitual sería responder que la probabilidad es 0.5. La primera intuición nos hace pensar que si el lado visible de la carta seleccionada es de color verde, descartaríamos una de las opciones iniciales (la carta que es roja por ambos lados). Esto nos hace pensar que nuestras probabilidades se centraría en una entre dos opciones, es decir, que al girar la carta o sea verde o sea roja.

   Pero este pensamiento inicial lo hacemos sin tomar en consideración que hay una condición previa y que la probabilidad no depende solo de la segunda acción sino que esta condicionada por la primera.

   Esto lo verificaremos posteriormente en el segundo inciso al hacer una simulación en Python para saber la verdadera probabilidad que resulto ser 2/3

b. **Realiza una simulación en Python para este problema, estimando la probabilidad tras realizar una cantidad elevada de simulaciones del problema. La función deberá tener como parámetro de entrada el n´umero de simulaciones n y como parámetro de salida una estimación de la probabilidad p. Para simular la extracción de las cartas y del lado visible tras la extracción puedes valerte de `random.randint`. Recuerda que durante la simulación sólo deberán contabilizarse los experimentos en los que la cara visible de la carta obtenida ha sido verde y, sobre estos, estimar la proporción de veces en las que la carta de la que procede ha resultado ser la que es verde por ambas caras.**

   Para corroborar la probabilidad de este problema, se implementó un algoritmo estocástico en Python que al simular la situación N cantidad de veces nos permite aproximarnos de mejor forma a la probabilidad buscada a medida que se incrementa el valor de N.

   A pesar de mostrar el código en este documento, el mismo se adjunta en un Jupyter Notebook que se encuentra anexo y donde se puede ejecutar el mismo.

```python
import random

```

```
numero_de_simulaciones = 100000 # Nro de simulaciones que se
    ejecutar n

def simulacion(numero_de_simulaciones):
    contador_de_exitos = 0
    contador_de_experimentos = 0

    # Definici n de las cartas
    cartas = [('roja', 'roja'), ('roja', 'verde'), ('verde', '
        verde')]

    for _ in range(numero_de_simulaciones):
        # Se elige una carta de las tres posibles
        carta_de_la_bolsa = cartas[random.randint(0,2)]

        # De la carta elegida se toma una de las caras
        cara_visible = carta_de_la_bolsa[random.randint(0,1)]

        if cara_visible == 'verde': # Si la cara es verde se
            cuenta el experimento
            contador_de_experimentos += 1
            if carta_de_la_bolsa == ('verde', 'verde'): # Si la
                carta es la verde, verde se cuenta como  xito
                contador_de_exitos += 1

    # Se imprime la probabilidad, relaci n de casos de exito
        entre casos contabilizados (cara visible verde)
    print(contador_de_exitos/contador_de_experimentos)

simulacion(numero_de_simulaciones)
```

El resultado de ejecutar este código y con 100.000 simulaciones resultó en una probabilidad de: 0.6639668826493881

c. **¿Son los resultados obtenidos consistentes con tu intuición inicial? De no ser el caso, ¿puedes conjeturar el valor correcto de la probabilidad? Una vez realizada, demuestra formalmente que el valor de la probabilidad conjeturada es el correcto y/o proporciona esquemáticamente la explicación de por qué se verifica.**

La simulación evidencia una probabilidad de 2/3 o 0,666666.

Generalmente por intuición descartamos información con la que ya contamos y en este caso es que hay un suceso inicial. Este es un típico ejemplo de Probabilidad Condicionada, donde tenemos que:

P(de que la carta sea la verde, verde) = P(vv) = 1/3, una de tres cartas posibles. P(de que el lado visible de la primera carta sea verde) = P(cv) = 3/6 = 1/2, tres opciones de 6 posibles. P(vv y cv).

Esto podemos esquematizarlo del siguiente modo. Dada la condición de que la cara mostrada ha sido verde, existe la misma probabilidad de 1/3 entre 3 opciones descritas en la tabla de abajo. Que sea la cara verde de la carta verde-roja o que sea algunas de las caras de la carta verde-verde. Al girar y ver la otra cara solo será verde en 2 de 3 casos, es decir 2/3.

Carta seleccionada	Cara mostrada	Probabilidad	La otra cara	Exito
verde-roja	verde	1/3	roja	No
verde-verde	verde 1	1/3	verde 2	Si
verde-verde	verde 2	1/3	verde 1	Si