

Introducción a la IA

Ejercicio 1

Defina con sus propias palabras: inteligencia, inteligencia artificial.

Inteligencia Artificial: La inteligencia artificial es el estudio sobre cómo hacer que las computadoras hagan tareas que, por el momento, los humanos hacen mejor. Esta definición corresponde al enfoque de *Sistemas que actúan como humanos*. Existen otros tres enfoques que pueden dar otras definiciones: *Sistemas que piensan como humanos*, *Sistemas que piensan racionalmente* y *Sistemas que actúan racionalmente*.

Inteligencia: Facultad que permite aprender, entender, razonar, tomar decisiones y formarse una idea determinada de la realidad.

Ejercicio 2

En realidad, los computadores no son inteligentes, hacen solamente lo que le dicen los programadores. ¿Es cierta esta aseveración, e implica a la primera?

Los programadores diseñan las computadoras de manera que puedan reaccionar a las instrucciones proporcionadas por los usuarios. Además, agregan datos y emplean algoritmos que, en conjunto, contribuyen a desarrollar un nivel de "inteligencia" en estas máquinas.

Ejercicio 3

Consulte en la literatura existente sobre la IA si alguna de las siguientes tareas se puede efectuar con computadores:

- **Jugar una partida de tenis de mesa (ping-pong) decentemente.**
- **Conducir un coche en el centro del Cairo.**
- **Comprar comestibles para una semana en el mercado.**
- **Comprar comestibles para una semana en la web.**
- **Descubrir y demostrar nuevos teoremas matemáticos.**
- **Escribir intencionadamente una historia divertida.**
- **Ofrecer asesoría legal competente en un área determinada.**
- **Traducir inglés hablado al sueco hablado en tiempo real.**
- **Realizar una operación de cirugía compleja.**

Jugar una partida de tenis de mesa (ping-pong) decentemente: Se han creado robots que pueden jugar al ping-pong, pero su nivel de habilidad puede variar y a menudo no son tan buenos como jugadores humanos profesionales.

Conducir un coche en el centro del Cairo: La conducción autónoma ha avanzado considerablemente en los últimos años, pero todavía enfrenta desafíos significativos, especialmente en entornos urbanos complejos como el centro de una gran ciudad como El Cairo. Empresas como Waymo y Tesla han desarrollado sistemas de conducción autónoma, pero la seguridad sigue siendo una preocupación importante.

Comprar comestibles para una semana en el mercado: La compra de comestibles en un mercado requiere la interacción física con productos y personas, lo que hace que esta tarea sea difícil de automatizar completamente con IA.

Comprar comestibles para una semana en la web: La compra de comestibles en línea es común y es una tarea que la IA puede facilitar, especialmente en términos de recomendaciones de productos y personalización de la experiencia de compra en línea.

Descubrir y demostrar nuevos teoremas matemáticos: La IA, en particular el aprendizaje automático simbólico y las redes neuronales, ha demostrado ser útil en la asistencia a matemáticos en la búsqueda de pruebas y teoremas, pero no ha llegado al punto de reemplazar completamente a los matemáticos en la creación de nuevos teoremas.

Escribir intencionadamente una historia divertida: La IA es capaz de generar texto y, a través del aprendizaje automático, puede generar historias que pueden ser entretenidas. Sin embargo, la capacidad de crear historias verdaderamente creativas y graciosas sigue siendo un desafío y a menudo depende de la calidad de los datos y el diseño del modelo.

Ofrecer asesoría legal competente en un área determinada: La IA ha avanzado en el campo legal en términos de análisis de documentos legales y búsqueda de información legal, pero aún no puede reemplazar a los abogados en la prestación de asesoramiento legal completo y competente, especialmente en áreas complejas.

Traducir inglés hablado al sueco hablado en tiempo real: Los sistemas de traducción automática han avanzado considerablemente, pero la traducción en tiempo real puede ser más desafiante debido a la complejidad de los idiomas y las diferencias culturales. Aun así, existen servicios de traducción automática en tiempo real que pueden ofrecer traducciones útiles.

Realizar una operación de cirugía compleja: La cirugía asistida por robot es una realidad y se utiliza en algunas cirugías complejas. Sin embargo, los cirujanos humanos todavía supervisan y controlan estas máquinas, ya que la precisión y la toma de decisiones son críticas en la cirugía.

Ejercicio 4

Explique sintéticamente las cuatro categorías que se vienen siguiendo respecto de la definición de inteligencia artificial.

Existen cuatro enfoques sobre la inteligencia artificial:

1. Sistemas que piensan como humanos
2. Sistemas que actúan como humanos
3. Sistemas que piensan racionalmente
4. Sistemas que actúan racionalmente

Los enfoques 1) y 3) se refieren a procesos mentales y al razonamiento, mientras que los enfoques 2) y 4) aluden a la conducta.

A su vez, los enfoques 1) y 2) miden el éxito en términos de la fidelidad en la forma de actuar de los humanos, en tanto que el 3) y 4) toman como referencia un concepto ideal de inteligencia, que llamaremos racionalidad. Un sistema es racional si hace «lo correcto», en función de su conocimiento.

El enfoque centrado en el comportamiento humano debe ser una ciencia empírica, que incluya hipótesis y confirmaciones mediante experimentos. El enfoque racional implica una combinación de matemáticas e ingeniería.

Agentes Inteligentes

Ejercicio 1

Defina con sus propias palabras los siguientes términos: agente, función de agente, programa de agente, racionalidad, autonomía, agente reactivo, agente basado en modelo, agente basado en objetivos, agente basado en utilidad, agente que aprende.

Agente: es un sistema o entidad que es capaz de percibir su entorno, procesar información y tomar decisiones para lograr sus objetivos o metas. Los agentes pueden ser físicos, como robots, o virtuales, como programas de software.

Función de Agente: describe la relación entre las percepciones del agente y las acciones que debe tomar en función de esas percepciones para lograr sus objetivos. En esencia, es una "regla" que guía el comportamiento del agente.

Programa de Agente: es el software o conjunto de algoritmos que implementa la función de agente. Es la parte ejecutable que permite que un agente realice acciones basadas en sus percepciones y objetivos.

Racionalidad: se refiere a la capacidad de un agente para tomar decisiones que maximicen el logro de sus objetivos. Un agente se considera racional cuando elige las acciones que tienen más probabilidades de llevarlo a sus metas.

Autonomía: se refiere a su capacidad para operar de manera independiente y tomar decisiones por sí mismo sin intervención externa constante. Los agentes autónomos son capaces de actuar en función de sus objetivos y percepciones sin necesidad de control constante.

Agente Reactivo: es un agente que toma decisiones basadas únicamente en la percepción actual de su entorno. No tiene una representación interna del mundo ni considera el pasado ni el futuro.

Agente Basado en Modelo: tiene una representación interna del mundo y puede utilizar ese modelo para planificar y tomar decisiones. Considera cómo evolucionará el entorno en el futuro.

Agente Basado en Objetivos: tiene metas claras y planifica sus acciones para alcanzar esas metas. Considera no solo el estado actual sino también el estado deseado.

Agente Basado en Utilidad: evalúa las acciones en función de su utilidad o valor esperado, teniendo en cuenta la incertidumbre y los posibles resultados. Intenta maximizar la utilidad esperada en lugar de simplemente alcanzar objetivos específicos.

Agente que Aprende: es aquel que mejora su rendimiento con el tiempo a través de la experiencia y la retroalimentación. Puede adaptarse y ajustar su comportamiento en función de la información que adquiere durante su funcionamiento.

Ejercicio 2

Identifique la descripción REAS que define el entorno de trabajo para cada uno de los siguientes agentes:

- a) Robot que juega al fútbol
- b) Agente para comprar libros en Internet
- c) Explorador autónomo de Marte
- d) Asistente matemático para la demostración de teoremas.

a) Robot que juega al fútbol:

Reglas (R): Las reglas podrían incluir estrategias de juego, tácticas para interactuar con otros jugadores y decisiones sobre cómo avanzar o defender.

Entorno (E): El entorno es el campo de fútbol, incluyendo su tamaño, las dimensiones de la portería, la ubicación de otros jugadores y la pelota.

Actuadores (A): Los actuadores son los componentes mecánicos y electrónicos del robot que le permiten moverse, patear la pelota y realizar acciones físicas.

Sensores (S): Los sensores incluyen cámaras, sensores de proximidad y otros dispositivos que permiten al robot percibir la ubicación de la pelota, la posición de los oponentes y el estado general del juego.

b) Agente para comprar libros en Internet:

Reglas (R): Las reglas pueden incluir pautas de compra, preferencias del usuario, límites de gasto y métodos de búsqueda de libros en línea.

Entorno (E): El entorno es el sitio web de comercio electrónico en el que se realizan las compras, incluyendo la interfaz de usuario, el catálogo de libros y las opciones de pago.

Actuadores (A): Los actuadores son las acciones que realiza el agente en el sitio web, como hacer clic en botones, completar formularios y realizar pagos.

Sensores (S): Los sensores incluyen la información disponible en el sitio web, como los detalles del libro, el precio actual y la disponibilidad.

c) Explorador autónomo de Marte:

Reglas (R): Las reglas pueden incluir instrucciones de navegación, decisiones sobre cuándo recopilar datos científicos y protocolos de comunicación con la Tierra.

Entorno (E): El entorno es la superficie de Marte, con sus condiciones geográficas y climáticas, así como la presencia de rocas, terreno y otros obstáculos.

Actuadores (A): Los actuadores son los mecanismos que permiten al explorador moverse, manipular instrumentos y realizar tareas específicas en Marte.

Sensores (S): Los sensores incluyen cámaras, instrumentos científicos y sistemas de navegación que permiten al explorador recopilar datos sobre el terreno, la atmósfera y otros aspectos del planeta.

d) Asistente matemático para la demostración de teoremas:

Reglas (R): Las reglas son los algoritmos y métodos matemáticos utilizados para demostrar teoremas, incluyendo estrategias de prueba y razonamiento lógico.

Entorno (E): El entorno es el campo de las matemáticas y la lógica formal, donde se plantean y se buscan soluciones a problemas matemáticos y teoremas.

Actuadores (A): Los actuadores pueden incluir la generación de pruebas formales, la simplificación de expresiones matemáticas y la presentación de resultados de manera comprensible.

Sensores (S): Los sensores son los datos matemáticos, ecuaciones y proposiciones que se presentan al asistente para su análisis y procesamiento.

Ejercicio 3

Dados los siguientes tipos de agente:

1. Robot que juega al fútbol
2. Agente para comprar libros en Internet
3. Explorador autónomo de Marte
4. Asistente matemático para la demostración de teoremas

Caracterice el medio de acuerdo con las propiedades de los entornos de trabajo:

- a) Totalmente observable vs. parcialmente observable
- b) Determinista vs. estocástico
- c) Episódico vs. secuencial
- d) Estático vs. dinámico
- e) Discreto vs. continuo

f) Agente individual vs. multiagente

1. Robot que juega al fútbol:

Totalmente observable: Parcialmente observable, ya que el robot no tiene una visión completa de todo el campo de fútbol en todo momento debido a su limitado campo de visión y a la constante acción y movimiento de otros jugadores.

Determinista: Estocástico, ya que las acciones de los jugadores adversarios y la trayectoria de la pelota pueden ser impredecibles.

Episódico: Secuencial, ya que el juego se desarrolla en una serie continua de eventos y acciones.

Estático: Dinámico, ya que el entorno cambia constantemente debido al movimiento de los jugadores y la pelota.

Discreto: Continuo, ya que las acciones del robot, como moverse y patear, pueden ser continuas en el tiempo.

Multiagente: Multiagente, ya que interactúa con otros jugadores y debe coordinarse con su propio equipo.

2. Agente para comprar libros en Internet:

Totalmente observable: Totalmente observable, ya que el agente tiene acceso completo a la información disponible en el sitio web de compra de libros.

Determinista: Determinista, ya que las acciones de búsqueda y compra en un sitio web suelen ser predecibles y controlables.

Episódico: Secuencial, ya que el proceso de búsqueda y compra de libros se realiza en una secuencia de pasos.

Estático: Estático, ya que el catálogo de libros y la interfaz del sitio web no cambian con frecuencia durante una sesión de compra.

Discreto: Discreto, ya que las acciones del agente, como hacer clic en botones y seleccionar opciones, son discretas.

Agente individual: Individual, ya que un usuario típico realiza compras en línea de forma individual.

3. Explorador autónomo de Marte:

Totalmente observable: Parcialmente observable, ya que la visibilidad del explorador puede estar limitada por la topografía de Marte, la distancia y las condiciones climáticas.

Determinista: Estocástico, ya que las condiciones en Marte, como tormentas de polvo, pueden ser impredecibles.

Episódico: Secuencial, ya que la exploración de Marte implica una serie de misiones y acciones en un período prolongado.

Estático: Dinámico, ya que el clima y las condiciones superficiales de Marte pueden cambiar con el tiempo.

Discreto: Discreto, ya que las acciones del explorador, como moverse y recopilar datos, son discretas.

Individual: Individual, ya que se refiere a un solo explorador autónomo.

4. Asistente matemático para la demostración de teoremas:

Totalmente observable: Totalmente observable, ya que el asistente tiene acceso completo a la información matemática proporcionada para demostrar teoremas.

Determinista: Determinista, ya que las reglas y axiomas de las matemáticas son consistentes y predecibles.

Episódico: Secuencial, ya que la demostración de teoremas implica una secuencia de pasos lógicos.

Estático: Estático, ya que las reglas matemáticas fundamentales no cambian durante la demostración.

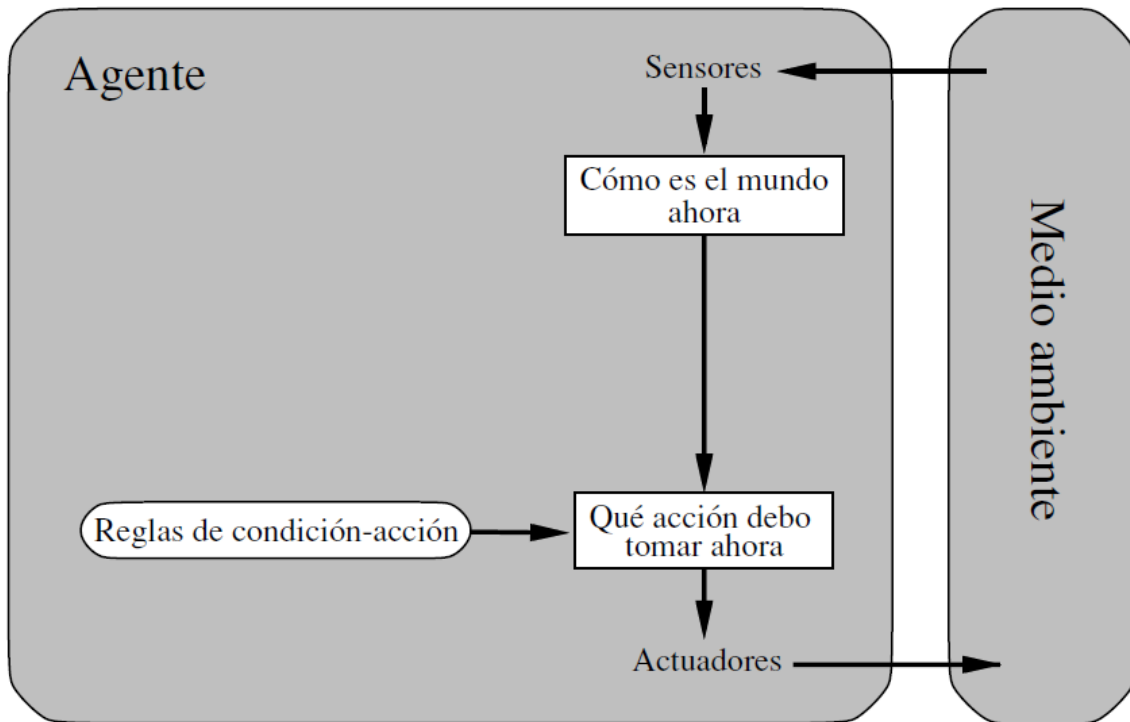
Discreto: Discreto, ya que las acciones del asistente, como aplicar reglas lógicas, son discretas.

Individual: Individual, ya que se refiere a un asistente matemático en particular que ayuda en la demostración de teoremas.

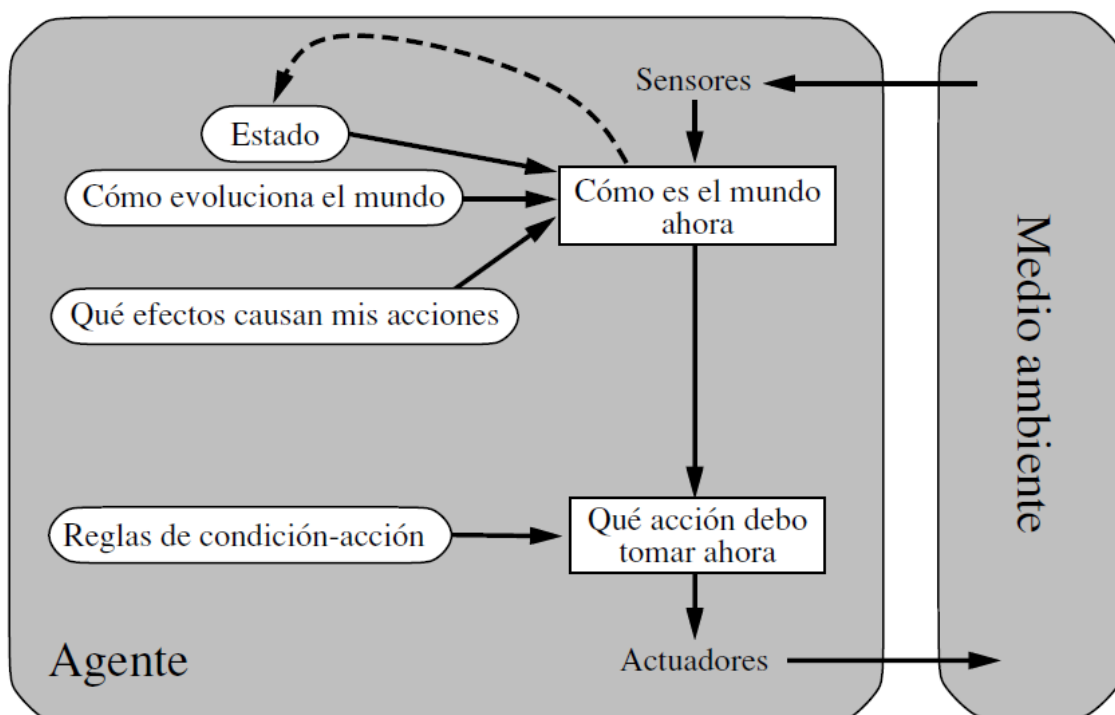
Ejercicio 4

Represente a través de diagramas esquemáticos los cuatro tipos de modelos de agentes.

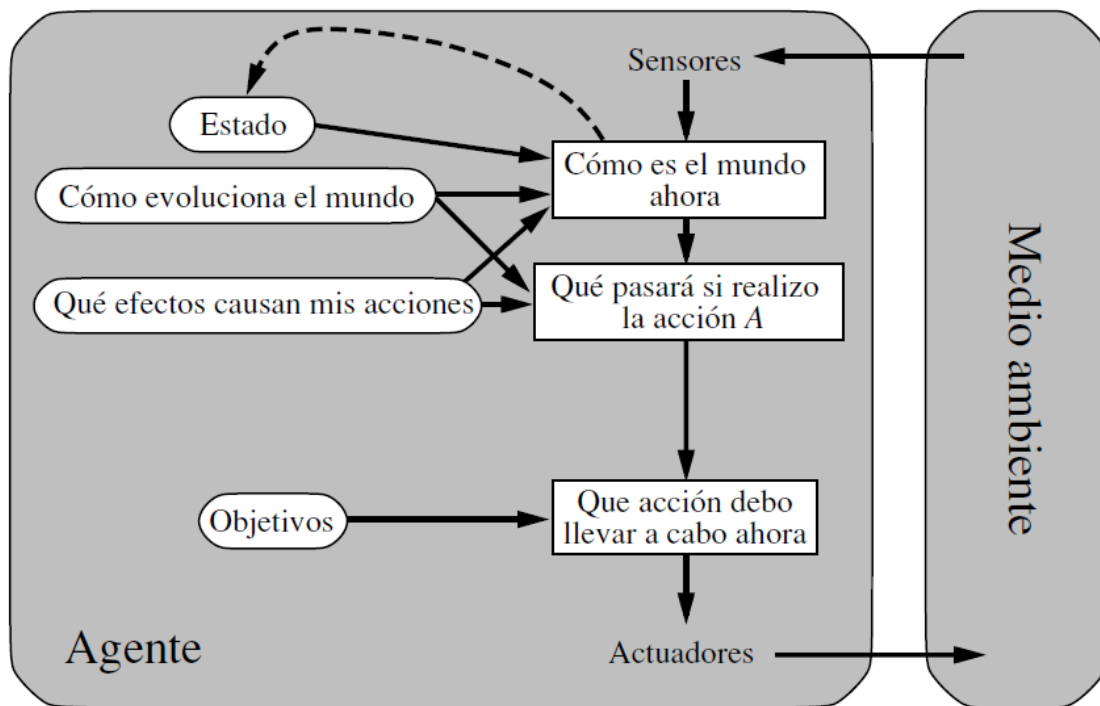
Agentes reactivos simples



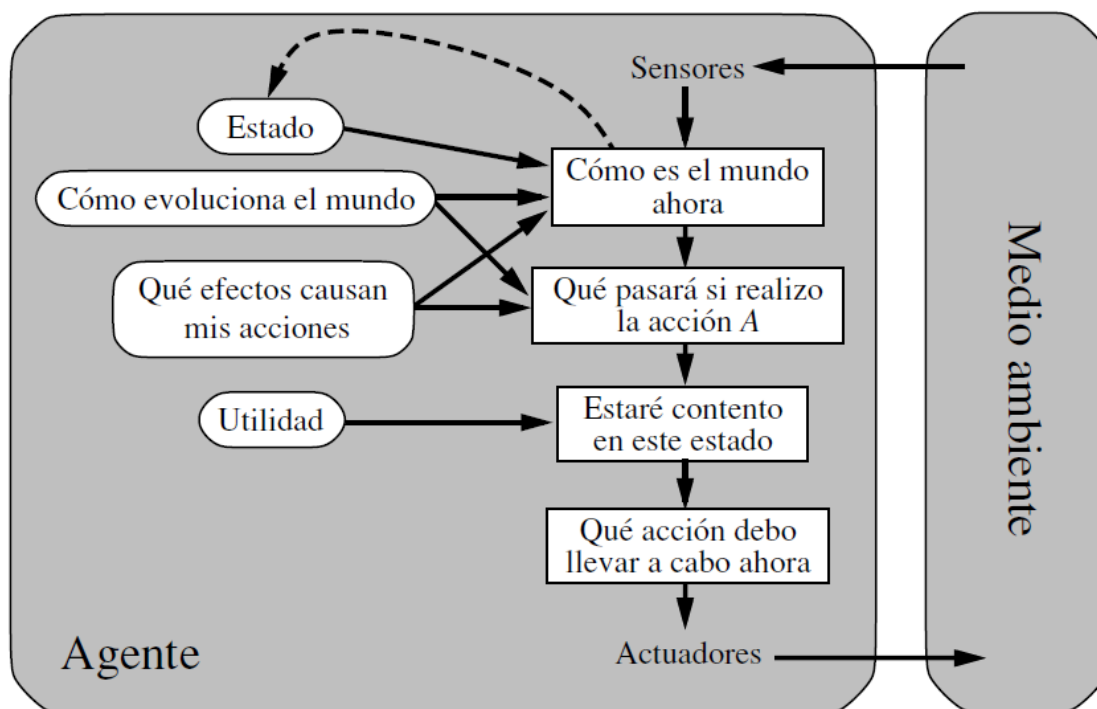
Agentes reactivos basados en modelos



Agentes basados en objetivos



Agentes basados en utilidad



Agentes Resolventes-Problemas

Ejercicio 1

Defina con sus propias palabras los siguientes términos: estado, espacio de estados, árbol de búsqueda, nodo de búsqueda, objetivo, acción, función sucesor, y factor de ramificación.

Estado: Un estado es una representación particular de una situación o condición en un problema o entorno. Puede ser cualquier configuración que describa dónde se encuentra el sistema en un momento dado.

Espacio de Estados: El espacio de estados es el conjunto completo de todos los posibles estados que un sistema o problema puede tomar. Es como un mapa que abarca todas las posibles situaciones que se pueden encontrar.

Árbol de Búsqueda: Un árbol de búsqueda es una estructura de datos utilizada en inteligencia artificial y resolución de problemas. Se utiliza para representar las diferentes etapas o estados en los que un agente puede estar mientras busca una solución.

Nodo de Búsqueda: Un nodo de búsqueda es un punto en el árbol de búsqueda que representa un estado específico en el proceso de búsqueda. Contiene información sobre el estado, su relación con otros estados y la ruta que llevó a ese estado.

Objetivo: El objetivo es la meta o condición que un agente o sistema intenta alcanzar en un problema. Es lo que se quiere lograr al final de la búsqueda.

Acción: Una acción es una operación o paso que un agente puede realizar para cambiar su estado en el espacio de estados. Las acciones son los movimientos o decisiones que se toman para avanzar hacia el objetivo.

Función Sucesor: La función sucesor es una función que describe cómo se transforma un estado actual en un nuevo estado después de aplicar una acción específica. Indica las posibles transiciones de un estado a otro.

Factor de Ramificación: El factor de ramificación se refiere al número promedio de nodos hijos en el árbol de búsqueda en un problema determinado. Indica cuántas opciones diferentes tiene un agente para tomar decisiones en cada estado. Un factor de ramificación alto significa que hay muchas posibles acciones, lo que puede hacer que la búsqueda sea más compleja.

Ejercicio 2

Explique por qué la formulación del problema debe seguir a la formulación del objetivo.

La formulación del problema debe seguir a la formulación del objetivo porque los objetivos proporcionan la base y la dirección para determinar qué problema se va a abordar y cómo se va a abordar (técnica a aplicar).

Ejercicio 3

Defina para cada uno de los problemas de juguete planteados: Estados, Estado inicial, Función sucesor, Espacio de Estados, Test objetivo, Costo del camino.

a) El mundo de la aspiradora

- **Estados**: existen dos localizaciones, la aspiradora puede estar en una de estas a la vez y cada una de ellas puede o no estar sucia (ambas sucias, ambas limpias, una sucia y la otra limpia).
- **Estado inicial**: cualquiera de los estados anteriormente descritos.
- **Función sucesor**: moverse a la izquierda, moverse a la derecha, aspirar. (tres acciones legales posibles)
- **Espacio de estados**: hay 8 posibles estados diferentes ($2 \times 2^2 = 8$)
- **Test objetivo**: comprobar que ambas localizaciones estén limpias.
- **Costo del camino**: número de pasos que lo componen, es constante el costo individual.

b) El 8-puzzle

- **Estados:** ubicación de cada una de las 8 fichas (y del espacio en blanco) dentro del tablero de 3x3.
- **Estado inicial:** cualquier estado posible puede ser el inicial
- **Función sucesor:** mover una ficha particular hacia arriba, abajo, izquierda o derecha dentro del tablero (cuatro acciones legales posibles)
- **Espacio de estados:** existen 9! estados posibles (aunque en algunos sitios dicen que es la mitad de esto)
- **Test objetivo:** comprobar si el estado actual de las piezas coincide con el estado objetivo (el cual suele ser ordenarlas en forma ascendente dejando la casilla vacía al inicio o al final)
- **Costo del camino:** número de pasos que lo componen, es constante el costo individual.

c) Las 8-reinas

- **Estados:** cualquier combinación de 0 a 8 damas en el tablero de ajedrez (64 casilleros).
- **Estado inicial:** cero damas en el tablero.
- **Función sucesor:** añadir una dama en algún casillero desocupado del tablero.
- **Espacio de estados:**
- **Test objetivo:** ocho damas en el tablero donde ninguna ataque a otra.
- **Costo del camino:** no es de interés ya que se busca una solución válida no una óptima.

Ejercicio 4

Supongamos que ACCIONES-LEGALES(s) denota el conjunto de acciones que son legales en el estado "s", y RESULTADO(a,s) denota el estado que resulta de la realización de una acción legal "a" para un estado "s". Defina: **Estados**, **Estado inicial**, **Función sucesor** en términos ACCIONES-LEGALES y RESULTADO.

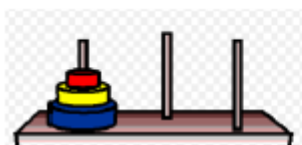
Estados: s

Estados inicial: Configuración inicial

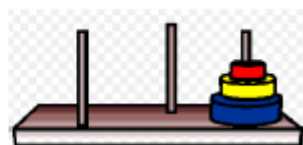
Función sucesor: Todos los estados posibles que se hayan generado aplicando la función sucesor.

Ejercicio 5

Dado el problema de las Torres de Hanoi, donde se deben colocar los tres discos en la torre C, partiendo de la torre A y valiéndose de la torre B para cumplir con el objetivo, defina: **Estados**, **Estado inicial**, **Función sucesor**.



Estado Inicial (1,2,3; ;)



Estado Final (; ;1,2,3)

Estados: Un estado se caracteriza por la disposición de los 3 discos en las tres torres en un momento dado.

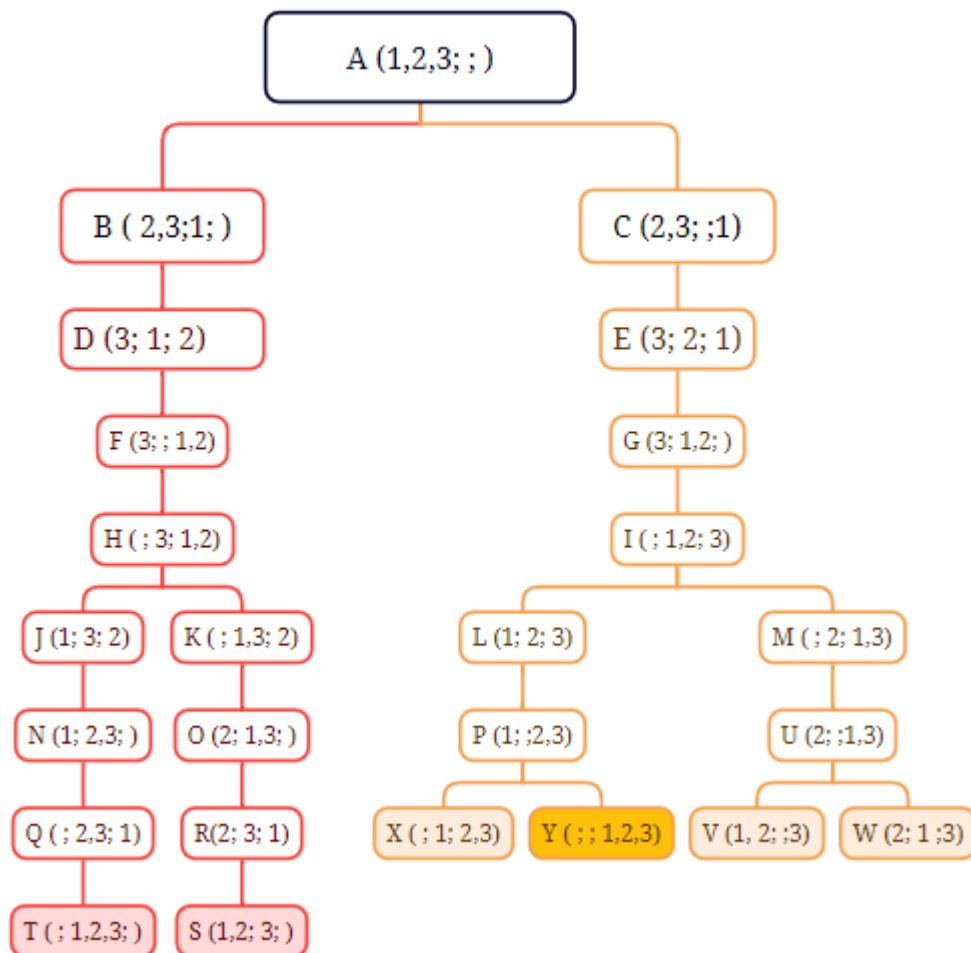
Estado inicial: Todos los discos sobre la torre A apilados de forma descendente y las torres B y C vacías.

Función sucesor: La función sucesor describe cómo se puede cambiar el estado actual para llegar a un nuevo estado. En las Torres de Hanoi, la función sucesor se define de la siguiente manera:

Se pueden mover discos individuales de la parte superior de una torre a la parte superior de otra torre, siempre y cuando el disco que se va a mover sea más pequeño que el disco en la parte superior de la torre de destino o si la torre de destino está vacía.

Ejercicio 6

Genere el árbol de búsqueda para el problema de las Torres de Hanoi.



Ejercicio 7

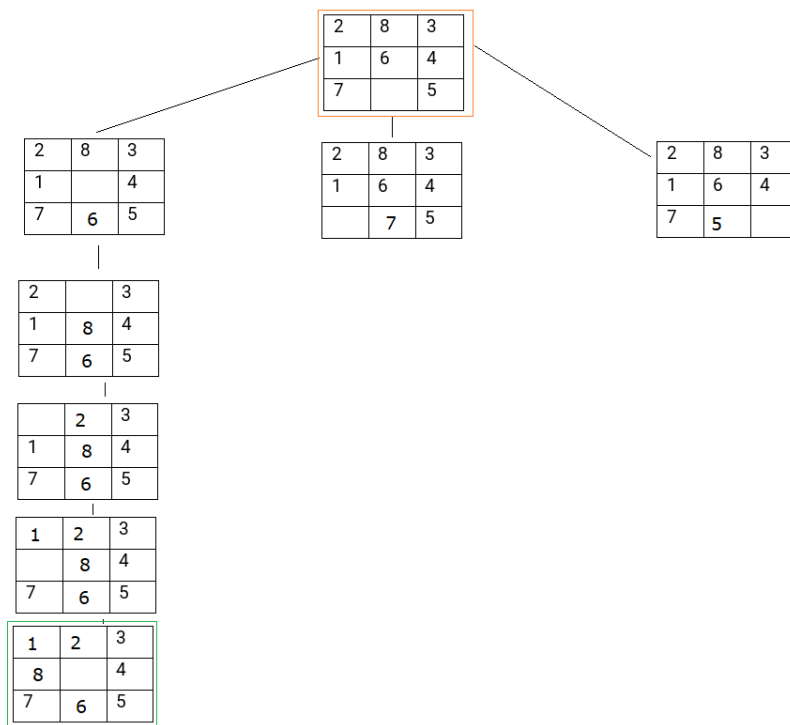
Genere el árbol de búsqueda para el problema del 8 –puzle. Acción: mover el blanco (izquierda, derecha, abajo, arriba).

2	8	3
1	6	4
7		5

Estado Inicial

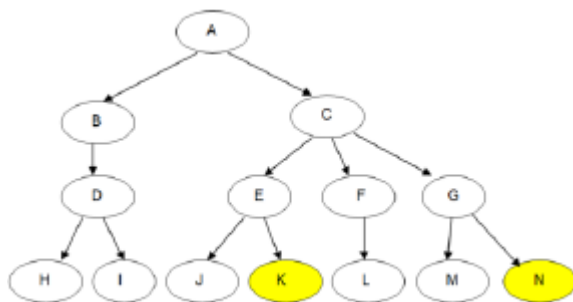
1	2	3
8		4
7	6	5

Estado Objetivo



Ejercicio 8

Si este árbol representa el árbol de búsqueda para el problema de las Torres de Hanoi, aplique la estrategia Búsqueda Primero en Anchura. Evalúe la estrategia en función de los criterios: Completitud, Optimización, Tiempo y Espacio. Considere que en el espacio de estados, el nodo **A** representa el estado inicial y los nodos **K** y **N** son estados objetivos.

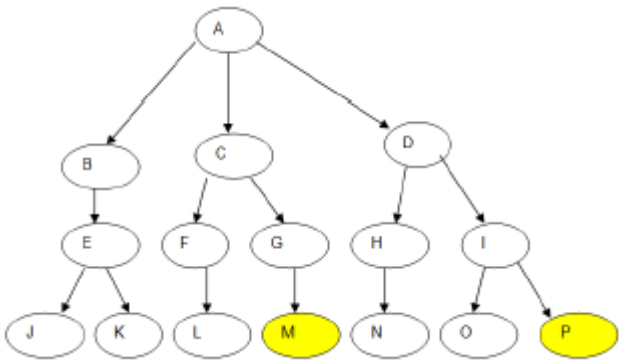


M	Lista Abierta	Lista Cerrada
-	A	-
M=A	B, C	A
M=B	C, D	A, B
M=C	D, E, F, G	A, B, C
M=D	E, F, G, H, I	A, B, C, D
M=E	F, G, H, I, J, K	A, B, C, D, E
M=F	G, H, I, J, K, L	A, B, C, D, E, F
M=G	H, I, J, K, L, M, N	A, B, C, D, E, F, G
M=H	I, J, K, L, M, N	A, B, C, D, E, F, G, H
M=I	J, K, L, M, N	A, B, C, D, E, F, G, H, I
M=J	K, L, M, N	A, B, C, D, E, F, G, H, I, J

M=K	L, M, N	A, B, C, D, E, F, G, H, I, J, K
-----	---------	---------------------------------

Ejercicio 9

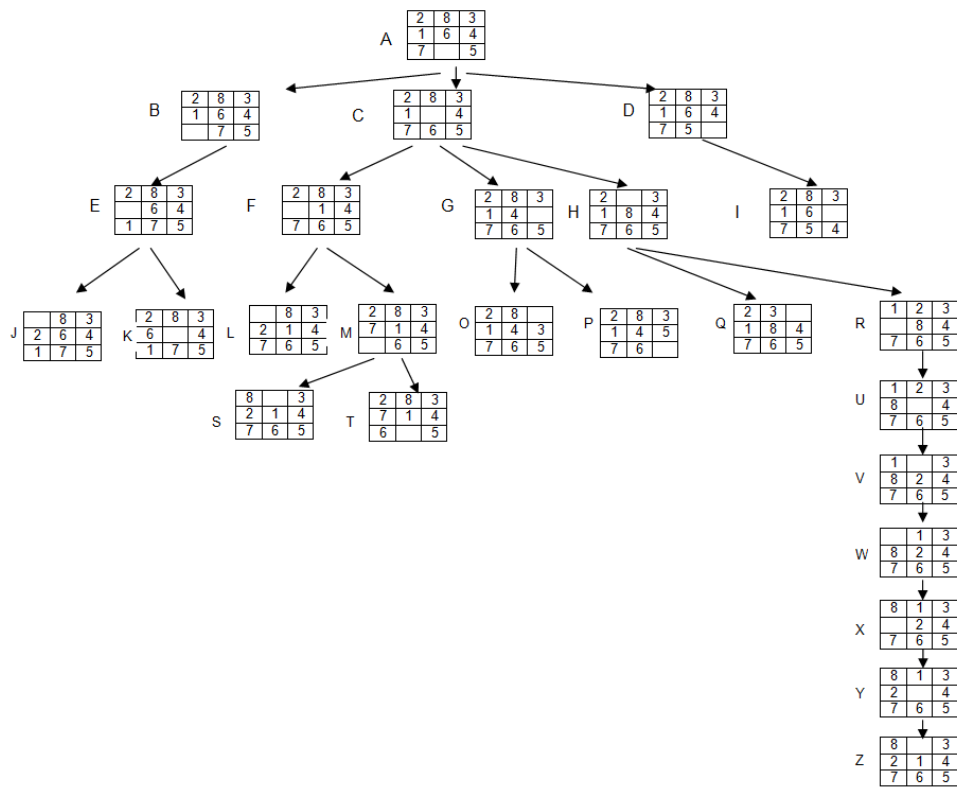
Si este árbol representa el árbol de búsqueda para el problema del 8-Puzzle, aplique la estrategia Búsqueda Primero en Profundidad. Evalúe la estrategia en función de los criterios: Completitud, Optimización, Tiempo y Espacio. Considere que en el espacio de estados, el nodo **A** representa el estado inicial y los nodos **M** y **P** son estados objetivos.



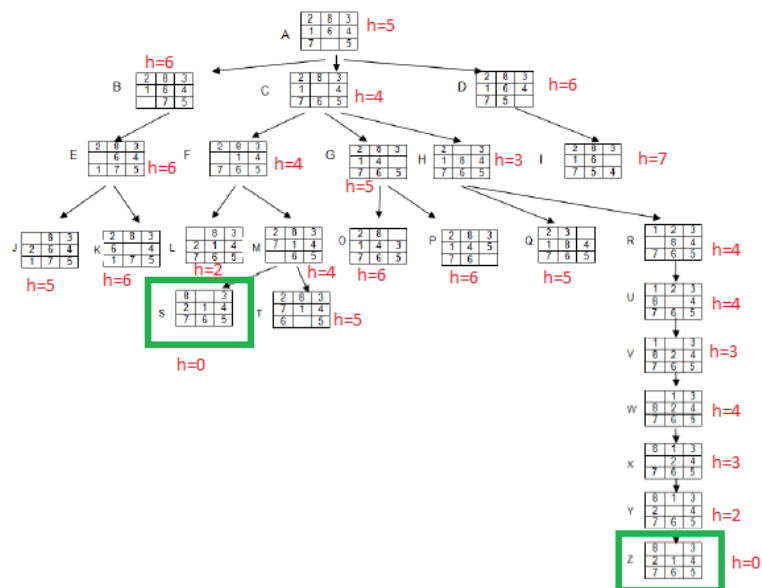
M	Lista Abierta	Lista Cerrada
-	A	-
M=A	B, C, D	A
M=B	E, C, D	A, B
M=E	J, K, C, D	A, B, E
M=J	K, C, D	A, B, E, J
M=K	C, D	A, B, E, J, K
M=C	F, G, D	A, B, E, J, K, C
M=F	L, G, D	A, B, E, J, K, C, F
M=L	G, D	A, B, E, J, K, C, F, L
M=G	M, D	A, B, E, J, K, C, F, L, G
M=M	D	A, B, E, J, K, C, F, L, G, M

Ejercicio 10

Dado el problema del 8-Puzzle, utilice la Heurística admisible h_1 = número de piezas mal ubicadas, asigne a cada nodo del siguiente árbol su $h(n)$ y resuelva aplicando el método de Búsqueda Primero el Mejor. Considere el nodo **A** como raíz del árbol y los nodos **S** y **Z** nodos objetivos.



POR CUESTIÓN DE ESPACIO NO SE CONTINUARÁ CON EL DESARROLLO DEL ÁRBOL, PERO EJEMPLIFICA LA EXPLOSIÓN DE CAMINOS ALTERNATIVOS PARA ARRIBAR A UNA SOLUCIÓN.



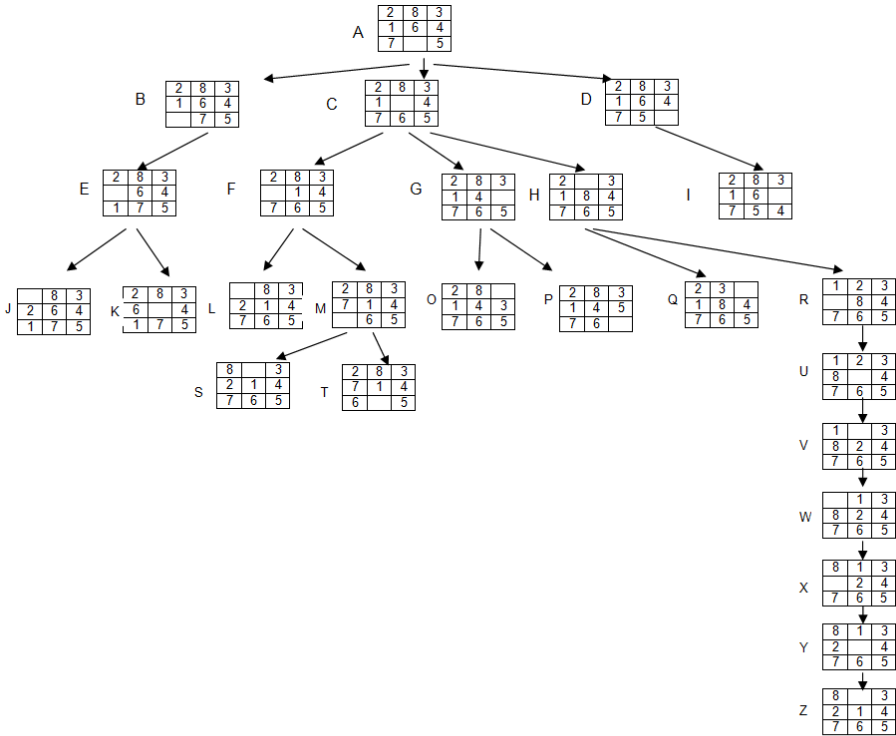
M	Lista Abierta	Lista Cerrada
-	A (5)	-
A (5)	B ^A (6), C ^A (4), D ^A (6)	A (5)
C ^A (4),	B ^A (6), D ^A (6), F ^C (4), G ^C (5), H ^C (3)	A (5) C ^A (4)
H ^C (3),	B ^A (6), D ^A (6), F ^C (4), G ^C (5), Q ^H (5), R ^H (4)	A (5) C ^A (4) H ^C (3),
F ^C (4),	B ^A (6), D ^A (6), G ^C (5), Q ^H (5), R ^H (4), L ^F (2), M ^F (4)	A (5) C ^A (4) H ^C (3), F ^C (4),
L ^F (2),	B ^A (6), D ^A (6), G ^C (5), Q ^H (5), R ^H (4), M ^F (4)	A (5) C ^A (4) H ^C (3), F ^C (4), L ^F (2)
R ^H (4)	B ^A (6), D ^A (6), G ^C (5), Q ^H (5), M ^F (4) U ^R (4)	A (5) C ^A (4) H ^C (3), F ^C (4), L ^F (2) R ^H (4)
M ^F (4)	B ^A (6), D ^A (6), G ^C (5), Q ^H (5), U ^R (4), S ^M (0), T ^M (5)	A (5) C ^A (4) H ^C (3), F ^C (4), L ^F (2) R ^H (4) M ^F (4)

S^M (0),	B^A (6), D^A (6), G^C (5), Q^H (5), U^R (4), T (5)	A (5) C^A (4) H^C (3), F^C (4), L^F (2) R^H (4) M^F (4) S^M (0)
------------	--	--

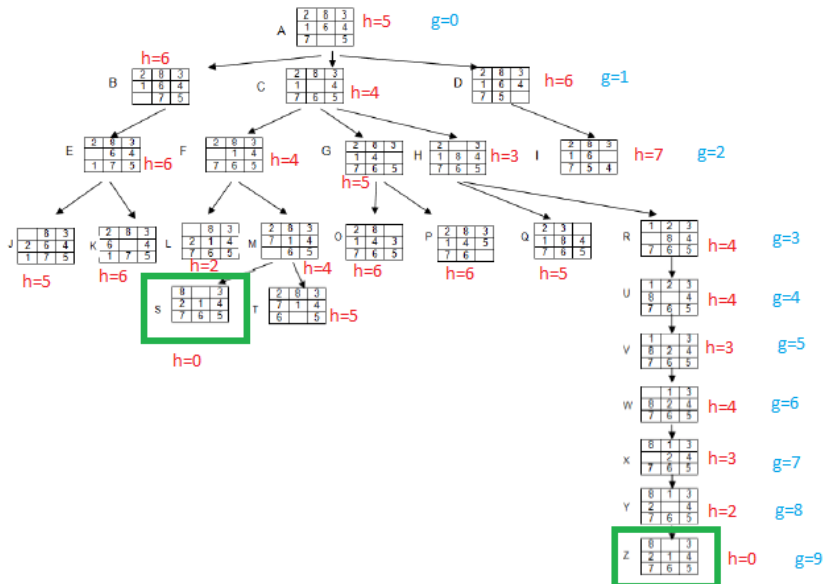
Camino: S^M , M^F , F^C , C^A , A

Ejercicio 11

Dado el problema del 8-Puzzle, utilice la Heurística admisible h_1 = número de piezas mal ubicadas, asigne a cada nodo del siguiente árbol su $h(n)$ y resuelva aplicando el método de Búsqueda A*. Considere el nodo **A** como raíz del árbol y los nodos **S y Z** nodos objetivos.



POR CUESTIÓN DE ESPACIO NO SE CONTINUARÁ CON EL DESARROLLO DEL ÁRBOL, PERO EJEMPLIFICA LA EXPLOSIÓN DE CAMINOS ALTERNATIVOS PARA ARRIBAR A UNA SOLUCIÓN.



M	Lista Abierta	Lista Cerrada
-	A (5+0)	-
A (5+0)	B^A (6+1), C^A (4+1), D^A (6+1)	A (5+0)
C^A (4+1)	B^A (6+1), D^A (6+1), F^C (4+2), G^C (5+2), H^C (3+2),	A (5+0) C^A (4+1)

H ^C (3+2)	B ^A (6+1), D ^A (6+1), F ^C (4+2), G ^C (5+2), Q ^H (5+3), R ^H (4+3),	A (5+0) C ^A (4+1) H ^C (3+2)
F ^C (4+2)	B ^A (6+1), D ^A (6+1), G ^C (5+2), Q ^H (5+3), R ^H (4+3), L ^F (2+3), M ^F (4+3),	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2)
L ^F (2+3)	B ^A (6+1), D ^A (6+1), G ^C (5+2), Q ^H (5+3), R ^H (4+3), M ^F (4+3),	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2) L ^F (2+3)
B ^A (6+1),	D ^A (6+1), G ^C (5+2), Q ^H (5+3), R ^H (4+3), M ^F (4+3), E (6+2),	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2) L ^F (2+3) B ^A (6+1)
D ^A (6+1)	G ^C (5+2), Q ^H (5+3), R ^H (4+3), M ^F (4+3), E (6+2), I (7+2),	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2) L ^F (2+3) B ^A (6+1) D ^A (6+1)
G ^C (5+2)	Q ^H (5+3), R ^H (4+3), M ^F (4+3), E (6+2), I (7+2), O (6+3), P (6+3),	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2) L ^F (2+3) B ^A (6+1) D ^A (6+1) G ^C (5+2)
R ^H (4+3)	Q ^H (5+3), M ^F (4+3), E (6+2), I (7+2), O ^G (6+3), P ^G (6+3), U ^R (4+4),	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2) L ^F (2+3) B ^A (6+1) D ^A (6+1) G ^C (5+2) R ^H (4+3)
M ^F (4+3)	Q ^H (5+3), M ^F (4+3), E (6+2), I (7+2), O ^G (6+3), P ^G (6+3), U ^R (4+4), S ^M (0+4) T ^M (5+4)	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2) L ^F (2+3) B ^A (6+1) D ^A (6+1) G ^C (5+2) R ^H (4+3) M ^F (4+3)
S ^L (0+4)	Q ^H (5+3), M ^F (4+3), E (6+2), I (7+2), O ^G (6+3), P ^G (6+3), U ^R (4+4), S ^M (0+4) T ^M (5+4)	A (5+0) C ^A (4+1) H ^C (3+2) F ^C (4+2) L ^F (2+3) B ^A (6+1) D ^A (6+1) G ^C (5+2) R ^H (4+3) M ^F (4+3)

Camino: S^M, M^F, F^C, C^A, A

Ejercicio 12

Escriba un programa que implemente un método de Búsqueda No Informado o Informado; puede seleccionar alguno de los enunciados propuestos en el libro de la Cátedra en el capítulo 3 o, si le resulta más conveniente, puede proponer usted una situación problemática.

```
from collections import deque

# Definir el grafo como un diccionario de adyacencia
grafo = {
    'A': ['B', 'C'],
    'B': ['D', 'E'],
    'C': ['F'],
    'D': [],
    'E': ['F'],
    'F': []
}

def busqueda_en_amplitud(grafo, inicio, objetivo):
    # Inicializar la cola para la búsqueda en amplitud
    cola = deque()
    cola.append(inicio)
    visitados = set()

    while cola:
        nodo = cola.popleft()
        if nodo == objetivo:
            return True # Se encontró el objetivo
        if nodo not in visitados:
            visitados.add(nodo)
            vecinos = grafo[nodo]
            for vecino in vecinos:
                cola.append(vecino)
    return False # No se encontró el objetivo

inicio = 'A'
```

```

objetivo = 'F'

if busqueda_en_amplitud(grafo, inicio, objetivo):
    print(f"Se encontró un camino de {inicio} a {objetivo} en el grafo.")
else:
    print(f"No se encontró un camino de {inicio} a {objetivo} en el grafo.")

```

Agentes Basados en Conocimiento

Ejercicio 1

Defina con sus propias palabras: Base de Conocimiento, Sentencia, Inferencia, Sintaxis de una sentencia, Semántica de una sentencia.

Base de Conocimiento: Almacén de información y reglas que un agente utiliza para razonar y tomar decisiones.

Sentencia: Una afirmación expresada en lenguaje formal que puede ser verdadera o falsa.

Inferencia: Proceso de razonamiento lógico para deducir nuevas conclusiones a partir de la información en la base de conocimiento.

Sintaxis de una Sentencia: Reglas de formato y estructura que definen cómo se construyen las sentencias en un lenguaje formal.

Semántica de una Sentencia: El significado o interpretación de una sentencia en el contexto del mundo real o del dominio que representa.

Ejercicio 2

Dada la siguiente inferencia en Lenguaje Natural, evaluar la validez de la misma por resolución:

"La casa de enfrente parece deshabitada desde hace unos días. Algunos diarios mojados yacen en el jardín. El pasto se ve crecido. Por lo tanto, la gente de esa casa debe estar de viaje".

Se puede representar los hechos aludidos mediante las cláusulas siguientes:

P = la gente de la casa de enfrente están en casa

Q = ellos juntan los diarios

R = ellos cortan el césped

Sentencia original	Forma clausal	Significado
P	P	La gente de la casa de enfrente están en casa
$\neg Q$	$\neg Q$	No juntan los diarios
$\neg R$	$\neg R$	No cortan el césped
$P \Rightarrow Q$	$\neg P \vee Q$	Si están en casa, juntan los diarios
$P \Rightarrow R$	$\neg P \vee R$	Si están en casa cortan el césped

$Q \wedge R \rightarrow P$ (implicancia)

$\neg Q \vee \neg R \vee P$ (forma clausal)

$\neg P \vee Q$	$\neg Q$
$\neg P$	P

[] \rightarrow cláusula vacía

$\neg P \vee R$	$\neg R$
$\neg P$	P

[] \rightarrow cláusula vacía

Ejercicio 3

A partir de la siguiente información y utilizando los siguientes predicados, aplique el método de resolución por contradicción, demostrando que $BC \neq \alpha$ probando que $BC \wedge \neg \alpha$ es insatisfacible derivando la cláusula vacía, para las consignas:

α = Personas mayores de 18 años.

Juan tiene 20 años, Pedro tiene 13 años, Ana tiene 10 años, Alejandra tiene 15 años, Jorge tiene 30 años, Luis tiene 28 años.

Predicados:

PERSONA (nombre, sexo, edad)

MAYOR_EDAD (nombre)

ANTECEDENTES CONSECUENTE

PERSONA(X,S,E) \wedge (E > 18) MAYOR_EDAD (X)

PERSONA(Juan,M,20)

\neg MAYOR_EDAD (X)

\neg PERSONA(X,S,E) \wedge (E > 18)

\neg PERSONA(Juan,M,20) \wedge (20 > 18)

[] Cláusula vacía.

Ejercicio 4

A partir de la siguiente información y utilizando los siguientes predicados, aplique el método de resolución por contradicción, demostrando que $BC \neq \alpha$ probando que $BC \wedge \neg \alpha$ es insatisfacible derivando la cláusula vacía, para las consignas:

- a) α = Son sospechosos del asesinato de Susana los hombres que tuvieron relación con Susana. ¿Es Juan sospechoso?
- b) α = Son sospechosas las mujeres que tuvieron relación con hombres que Susana conocía. ¿Es sospechosa Bárbara?

Información:

Bárbara es amiga de Juan, Bárbara es amiga de Roberto, Bárbara es amiga de María, Susana es amiga de Juan, Susana es amiga de Pedro.

Predicados:

PERSONA(nombre,sexo)

AMISTAD(mujer,hombre)

SOSPECHOSO(nombre)

Hechos: PERSONA(Susana,Femenino)
PERSONA(Bárbara,Femenino)
PERSONA(María,Femenino)
PERSONA(Juan,Masculino)
PERSONA(Roberto,Masculino)
PERSONA(Pedro,Masculino)
AMISTAD(Bárbara,Juan)
AMISTAD(Bárbara,Roberto)
AMISTAD(Susana,Juan)
AMISTAD(Susana,Pedro)

PERSONA(X,F) AMISTAD(X,M) PERSONA(Y,M) AMISTAD(S,Y) PERSONA(S,F) \rightarrow SOSPECHOSO(X)

RESOLUCIÓN POR CONTRADICCIÓN:

	Sospechoso(X)
PERSONA(Susana, Femenino)	\negPERSONA(X,F) \vee \negAMISTAD(X,Y) \vee \negPERSONA(Y,M) \vee \negAMISTAD(Susana,Y) \vee \negPERSONA(Susana,Femenino)
AMISTAD(Susana, Juan)	\negPERSONA(X,F) \vee \negAMISTAD(X,Juan) \vee \negPERSONA(Juan,M) \vee \negAMISTAD(Susana,Juan)
PERSONA(Juan, Masculino)	\negPERSONA(X,F) \vee \negAMISTAD(X,Juan) \vee \negPERSONA(Juan,Masculino)
AMISTAD(Bárbara, Juan)	\negPERSONA(Bárbara,F) \vee \negAMISTAD(Bárbara,Juan)
PERSONA(Bárbara, Femenino)	\negPERSONA(Bárbara,Femenino)
	[] Cláusula vacía.

Ejercicio 5

A partir de la siguiente información y utilizando los siguientes predicados, aplique el método de resolución por contradicción, demostrando que $BC \neq \alpha$ probando que $BC \wedge \neg \alpha$ es insatisfacible derivando la cláusula vacía, para la consigna: α = ¿Es Miguel un socio escalador de montañas?

Antonio y Miguel son socios de un club alpino

Todo socio que no es un esquiador, es un escalador de montañas

A los escaladores de montañas no les gusta la lluvia

Si a alguien no le gusta la nieve, no es un esquiador

A Miguel le disgusta todo lo que a Antonio le gusta y le gusta todo lo que a Antonio le disgusta.

A Antonio le gusta la nieve y la lluvia

Predicados:

SOCIO(nombre)

ESCALADOR(X) \leftarrow SOCIO(X), \neg ESQUIADOR(X), \neg GUSTA_LLUVIA(X)

GUSTA_LLUVIA(nombre)

GUSTA_NIEVE(nombre)

\neg ESQUIADOR(X) \leftarrow \neg GUSTA_NIEVE(X)

Hechos:

SOCIO(ANTONIO), SOCIO(MIGUEL)

GUSTA_LLUVIA(ANTONIO), GUSTA_NIEVE(MIGUEL),

\neg GUSTA_LLUVIA(MIGUEL), \neg GUSTA_NIEVE(ANTONIO)

Si no le gusta la nieve, no es esquiador: \neg ESQUIADOR(X) \leftarrow \neg GUSTA_NIEVE(X)

	\neg ESCALADOR(X)
SOCIO(MIGUEL)	\neg SOCIO(X) \vee \neg (\neg ESQUIADOR(X)) \vee \neg (\neg GUSTA_LLUVIA(X))
SOCIO(MIGUEL)	\negSOCIO(MIGUEL) \vee ESQUIADOR(MIGUEL) \vee GUSTA_LLUVIA(MIGUEL)
\negGUSTA_LLUVIA(MIGUEL)	ESQUIADOR(MIGUEL) \vee GUSTA_LLUVIA(MIGUEL)
\neg GUSTA_NIEVE(X) \vee \neg ESQUIADOR(X)	ESQUIADOR(MIGUEL)
\neg GUSTA_NIEVE(MIGUEL) \vee \neg ESQUIADOR(MIGUEL)	ESQUIADOR(MIGUEL)
\negGUSTA_NIEVE(MIGUEL)	GUSTA_NIEVE(MIGUEL)
	[] Cláusula vacía.

Planificación

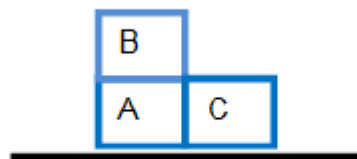
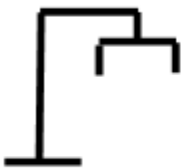
Dada la siguiente configuración de bloques y los operadores que se muestran a continuación, mostrar la evolución de la pila de objetivos para poder llegar desde el estado inicial al estado final.

- APILAR(x,y)
P: despejado(y) ^ agarrado(x)
B: despejado(y) ^ agarrado(x)
A: brazolibre ^ sobre(x,y)
- DESAPILAR(x,y)
P: sobre(x,y) ^ despejado(x) ^ brazolibre()
B: sobre(x,y) ^ brazolibre()
A: agarrado(x) ^ despejado(y)
- TOMAR(x)
P: despejado(x) ^ sobrelamesa(x) ^ brazolibre()
B: sobrelamesa(x) ^ brazolibre()
A: agarrado(x)
- BAJAR(x)
P: agarrado(x)
B: agarrado(x)
A: sobrelamesa(x) ^ brazolibre()

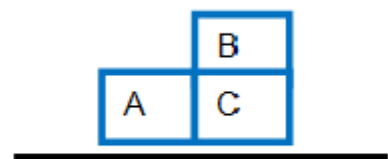


Ejercicio 1

1)



Estado Inicial



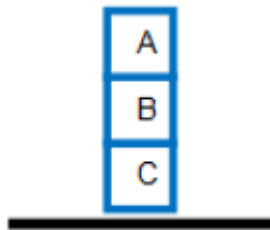
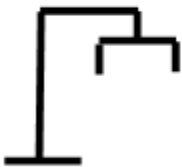
Estado Objetivo

Estado Actual		
Sobrelamesa(A) Sobrelamesa(C) Despejado(B) Despejado(C) Brazolibre() Sobre(B, A)	✓ despejado(B) ✓ sobrelamesa(A) ✓ sobrelamesa(C) despejado(A) sobre(B,C) despejado(B)^ despejado(A)^ sobrelamesa(A)^sobrelamesa(C), sobre(B,C)^brazolibre()	
Sobrelamesa(A) Sobrelamesa(C) Despejado(B) Despejado(C) Brazolibre() Sobre(B, A) Agarrado(B) Despejado(A)	✓ Sobre (B,A) ✓ despejado(B) ✓ sobrelamesa(A) ✓ Sobre (B,A) ^ despejado(B) ^ sobrelamesa(A) DESAPILAR (B,A) sobre(B,C) despejado(B)^ despejado(A)^ sobrelamesa(A)^sobrelamesa(C), sobre(B,C)^brazolibre()	DESAPILAR(B,A)

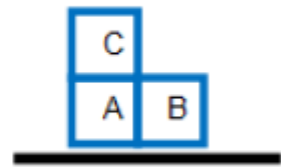
Sobrelamesa(A) Sobrelamesa(C) Despejado(B) Despejado(C) Agarrado(B) Despejado(A) Brazolibre() Sobre(B,C)	✓ Despejado (C) ✓ Agarrado(B) ✓ Despejado(C) ^ agarrado (B) APILAR(B,C) ✓ despejado(B)^ despejado(A)^ sobrelamesa(A)^sobrelamesa(C), sobre(B,C)^brazolibre()	DESAPILAR(B,A) APILAR(B,C)

Ejercicio 2

2)



Estado Inicial



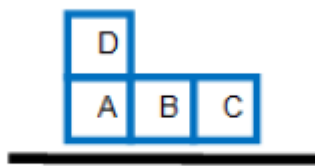
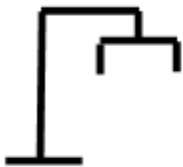
Estado Objetivo

Estado Actual		
Sobrelamesa(C) Despejado(A) Brazolibre() Sobre(A, B) Sobre(B, C)	✓ brazoLibre() despejado(B) despejado(C) sobrelamesa(A) sobrelamesa(B) sobre(C,A) despejado(B)^ despejado(C)^ sobrelamesa(A)^sobrelamesa(B), sobre(C,A)^brazolibre()	
Sobrelamesa(C) Despejado(A) Brazolibre() Sobre(A, B) Sobre(B, C) Agarrado(A) Despejado(B)	✓ Brazolibre() ✓ Despejado(A) ✓ Sobre(A,B) ✓ Brazolibre() ^ despejado (A) ^ sobre (A,B) Desapilar (A,B) despejado(C) sobrelamesa(A) sobrelamesa(B)	Desapilar (A,B)

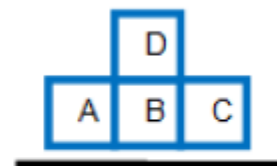
	sobre(C,A) despejado(B)^ despejado(C)^ sobrelamesa(A)^sobrelamesa(B), sobre(C,A)^brazolibre()	
Sobrelamesa(C) Despejado(A) Sobre(B, C) Agarrado(A) Despejado(B)	✓ Sobre(B,C) Brazolibre() Despejado(B) Sobre(B,C) ^ despejado (B) ^ brazoLibre() Desapilar (B,C) sobrelamesa(A) sobrelamesa(B) sobre(C,A) despejado(B)^ despejado(C)^ sobrelamesa(A)^sobrelamesa(B), sobre(C,A)^brazolibre()	Desapilar (A,B)
Sobrelamesa(C) Despejado(A) Sobre(B, C) Agarrado(A) Despejado(B) Sobrelamesa(A) Brazolibre() Agarrado(B) Despejado(C)	✓ Agarrado(A) Bajar (A) ✓ Despejado(B) ✓ Sobre(B,C) ^ despejado (B) ^ brazoLibre() Desapilar (B,C) ✓ sobrelamesa(A) sobrelamesa(B) sobre(C,A) despejado(B)^ despejado(C)^ sobrelamesa(A)^sobrelamesa(B), sobre(C,A)^brazolibre()	Desapilar(A,B) Bajar(A) Desapilar (B,C)
Sobrelamesa(C) Despejado(A) Despejado(B) Sobrelamesa(A) Agarrado(B) Despejado(C) brazoLibre() sobrelamesa(B)	✓ Agarrado(B) Bajar(B) sobre(C,A) despejado(B)^ despejado(C)^ sobrelamesa(A)^sobrelamesa(B), sobre(C,A)^brazolibre()	Desapilar(A,B) Bajar(A) Desapilar (B,C) Bajar (B)
Sobrelamesa(C) Despejado(A) Despejado(B) Sobrelamesa(A) Despejado(C)	✓ Despejado(C) Agarrado(C) Despejado(A) ^ agarrado (C) Apilar (C,A)	Desapilar(A,B) Bajar(A) Desapilar (B,C) Bajar (B)

brazoLibre() sobrelamesa(B)	despejado(B)^ despejado(C)^ sobrelamesa(A)^sobrelamesa(B), sobre(C,A)^brazolibre()	
Sobrelamesa(C) Despejado(A) Despejado(B) Sobrelamesa(A) Despejado(C) brazoLibre() sobrelamesa(B) agarrado(C) brazoLibre() sobre(C,A)	<ul style="list-style-type: none"> ✓ Despejado(C) ✓ Sobrelamesa(C) ✓ Brazolibre() ✓ Despejado(C)^sobrelamesa(C) ^brazolibre() Tomar(C) <ul style="list-style-type: none"> ✓ Despejado(A) ^ agarrado (C) Apilar (C,A) <ul style="list-style-type: none"> ✓ despejado(B)^ despejado(C)^ sobrelamesa(A)^sobrelamesa(B), sobre(C,A)^brazolibre() 	Desapilar(A,B) Bajar(A) Desapilar (B,C) Bajar (B) Tomar(C) Apilar(C,A)

Ejercicio 3



Estado Inicial



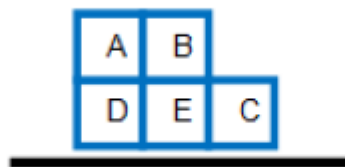
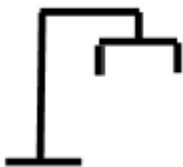
Estado Objetivo

Estado Actual		
Sobrelamesa(A) Sobrelamesa(B) Sobrelamesa(C) Despejado(D) Despejado(B) Despejado(C) Brazolibre() Sobre(D, A)	<ul style="list-style-type: none"> ✓ Sobrelamesa(A) ✓ Sobrelamesa(B) ✓ Sobrelamesa(C) ✓ Despejado(C) ✓ Despejado(D) Despejado (A) Sobre(D, B) Sobre(D, B) ^ Despejado (A) ^ Despejado (D) ^ Despejado (C) ^ Sobrelamesa (A) ^ Sobrelamesa (B) ^ Sobrelamesa (C) ^ brazolibre()	
Sobrelamesa(A) Sobrelamesa(B) Sobrelamesa(C) Despejado(D)	<ul style="list-style-type: none"> ✓ Sobre (D,A) ✓ despejado (D) ✓ brazolibre() ✓ Sobre (D,A) ^ despejado (D) ^ brazolibre() 	DESAPILAR (D,A)

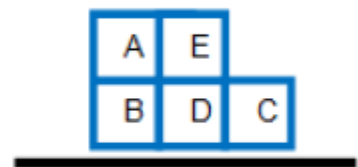
Despejado(B) Despejado(C) Brazolibre() Sobre(D, A) Agarrado(D) Despejado(A)	DESAPILAR (D,A) Sobre(D, B) Sobre(D, B) ^ Despejado (A) ^ Despejado (D) ^ Despejado (C) ^ Sobrelamesa (A) ^ Sobrelamesa (B) ^ Sobrelamesa (C) ^ brazolibre()	
Sobrelamesa(A) Sobrelamesa(B) Sobrelamesa(C) Despejado(D) Despejado(B) Despejado(C) Agarrado(D) Despejado(A) Brazolibre() Sobre(D,B)	✓ Despejado(B) ✓ agarrado (D) ✓ Despejado(B) ^ agarrado (D) APILAR(D, B) ✓ Sobre(D, B) ^ Despejado (A) ^ Despejado (D) ^ Despejado (C) ^ Sobrelamesa (A) ^ Sobrelamesa (B) ^ Sobrelamesa (C) ^ brazolibre()	APILAR(D,B)

Ejercicio 4

4)



Estado Inicial



Estado Objetivo

Estado Actual		
Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Brazolibre() Sobre(A, D)	✓ Despejado (C) ✓ Sobrelamesa (C) ✓ brazolibre() ✓ Despejado (A) Despejado (E) Sobrelamesa (D) Sobrelamesa (B) Sobre(E, D) Sobre(A, B)	

Sobre(B, E)	$\text{Sobre(A, B)} \wedge \text{Sobre(E, D)} \wedge \text{Despejado (A)} \wedge$ $\text{Despejado (E)} \wedge \text{Despejado (C)} \wedge$ $\text{Sobrelamesa (B)} \wedge \text{Sobrelamesa (D)} \wedge$ $\text{Sobrelamesa (C)} \wedge \text{brazolibre()}$	
Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Brazolibre() Sobre(A, D) Sobre(B, E) Agarrado(B) Despejado (E)	<ul style="list-style-type: none"> ✓ Brazolibre() ✓ Despejado(B) ✓ Sobre(B,E) ✓ $\text{Sobre(B,E)} \wedge \text{despejado (B)} \wedge$ brazolibre() DESAPILAR (B, E) <ul style="list-style-type: none"> ✓ Sobrelamesa (D) Sobrelamesa (B) Sobre(E, D) Sobre(A, B) $\text{Sobre(A, B)} \wedge \text{Sobre(E, D)} \wedge \text{Despejado (A)} \wedge$ $\text{Despejado (E)} \wedge \text{Despejado (C)} \wedge$ $\text{Sobrelamesa (B)} \wedge \text{Sobrelamesa (D)} \wedge$ $\text{Sobrelamesa (C)} \wedge \text{brazolibre()}$	DESAPILAR (B, E)
Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Sobre(A, D) Agarrado(B) Despejado (E) Sobrelamesa(B) Brazolibre()	<ul style="list-style-type: none"> ✓ Agarrado (B) BAJAR (B) Sobre(E, D) Sobre(A, B) $\text{Sobre(A, B)} \wedge \text{Sobre(E, D)} \wedge \text{Despejado (A)} \wedge$ $\text{Despejado (E)} \wedge \text{Despejado (C)} \wedge$ $\text{Sobrelamesa (B)} \wedge \text{Sobrelamesa (D)} \wedge$ $\text{Sobrelamesa (C)} \wedge \text{brazolibre()}$	DESAPILAR (B, E) BAJAR(B)
Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Sobre(A, D) Despejado (E)	Despejado(D) Agarrado(E) Despejado(D)^agarrado(E) APILAR(E,D) Sobre(A, B) $\text{Sobre(A, B)} \wedge \text{Sobre(E, D)} \wedge \text{Despejado (A)} \wedge$ $\text{Despejado (E)} \wedge \text{Despejado (C)} \wedge$ $\text{Sobrelamesa (B)} \wedge \text{Sobrelamesa (D)} \wedge$ $\text{Sobrelamesa (C)} \wedge \text{brazolibre()}$	DESAPILAR (B, E) BAJAR(B)

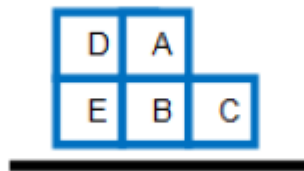
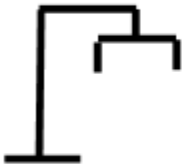
Sobrelamesa(B) Brazolibre()		
Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Sobre(A, D) Despejado (E) Sobrelamesa(B) Brazolibre() Agarrado(A) Despejado(D)	✓ Despejado(A) ✓ Sobre(A,D) ✓ Brazolibre() ✓ Despejado(A)^brazolibre()^sobre(A,D) DESAPILAR (A,D) Agarrado(E) Despejado(D)^agarrado(E) APILAR(E,D) Sobre(A, B) Sobre(A, B) ^ Sobre(E, D) ^ Despejado (A) ^ Despejado (E) ^ Despejado (C) ^ Sobrelamesa (B) ^ Sobrelamesa (D) ^ Sobrelamesa (C) ^ brazolibre()	DESAPILAR (B, E) BAJAR(B) DESAPILAR(A,D)
Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Despejado (E) Sobrelamesa(B) Agarrado(A) Despejado(D)	✓ Despejado(E) ✓ Sobrelamesa(E) Brazolibre() Despejado(E) ^ sobrelamesa(E) ^brazolibre() TOMAR (E) Despejado(D)^agarrado(E) APILAR(E,D) Sobre(A, B) Sobre(A, B) ^ Sobre(E, D) ^ Despejado (A) ^ Despejado (E) ^ Despejado (C) ^ Sobrelamesa (B) ^ Sobrelamesa (D) ^ Sobrelamesa (C) ^ brazolibre()	DESAPILAR (B, E) BAJAR(B) DESAPILAR(A,D)
Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Despejado (E) Sobrelamesa(B) Agarrado(A) Despejado(D) Sobrelamesa(A) Brazolibre()	✓ Agarrado(A) BAJAR(A) Despejado(E) ^ sobrelamesa(E) ^brazolibre() TOMAR (E) Despejado(D)^agarrado(E) APILAR(E,D) Sobre(A, B) Sobre(A, B) ^ Sobre(E, D) ^ Despejado (A) ^ Despejado (E) ^ Despejado (C) ^ Sobrelamesa (B) ^ Sobrelamesa (D) ^ Sobrelamesa (C) ^ brazolibre()	DESAPILAR (B, E) BAJAR(B) DESAPILAR(A,D) BAJAR(A)

Sobrelamesa(D) Sobrelamesa(E) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Despejado (E) Sobrelamesa(B) Despejado(D) Sobrelamesa(A) Brazolibre() Agarrado(E)	✓ Despejado(E) ^ sobrelamesa(E) ^brazolibre() TOMAR (E) Despejado(D)^agarrado(E) APILAR(E,D) Sobre(A, B) Sobre(A, B) ^ Sobre(E, D) ^ Despejado (A) ^ Despejado (E) ^ Despejado (C) ^ Sobrelamesa (B) ^ Sobrelamesa (D) ^ Sobrelamesa (C) ^ brazolibre()	DESAPILAR (B, E) BAJAR(B) DESAPILAR(A,D) BAJAR(A) TOMAR(E)
Sobrelamesa(D) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Despejado (E) Sobrelamesa(B) Despejado(D) Sobrelamesa(A) Agarrado(E) Brazolibre() Sobre(E,D)	✓ Despejado(D)^agarrado(E) APILAR(E,D) Sobre(A, B) Sobre(A, B) ^ Sobre(E, D) ^ Despejado (A) ^ Despejado (E) ^ Despejado (C) ^ Sobrelamesa (B) ^ Sobrelamesa (D) ^ Sobrelamesa (C) ^ brazolibre()	DESAPILAR (B, E) BAJAR(B) DESAPILAR(A,D) BAJAR(A) TOMAR(E) APILAR(E,D)
Sobrelamesa(D) Sobrelamesa(C) Despejado(A) Despejado(B) Despejado(C) Despejado (E) Sobrelamesa(B) Sobrelamesa(A) Brazolibre() Sobre(E,D)	✓ Despejado(B) Agarrado(A) Agarrado(A) ^ despejado (B) APILAR(A,B) Sobre(A, B) ^ Sobre(E, D) ^ Despejado (A) ^ Despejado (E) ^ Despejado (C) ^ Sobrelamesa (B) ^ Sobrelamesa (D) ^ Sobrelamesa (C) ^ brazolibre()	DESAPILAR (B, E) BAJAR(B) DESAPILAR(A,D) BAJAR(A) TOMAR(E) APILAR(E,D)
Sobrelamesa(D) Sobrelamesa(C) Despejado(A) Despejado(B)	Sobrelamesa(A) Brazolibre() Sobrelamesa(A) ^ brazolibre() TOMAR(A)	DESAPILAR (B, E) BAJAR(B) DESAPILAR(A,D) BAJAR(A)

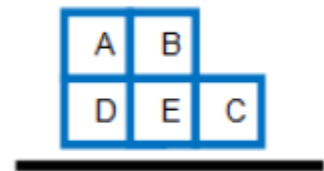
Despejado(C)	Agarrado(A) ^ despejado (B)	TOMAR(E)
Despejado (E)	APILAR(A,B)	APILAR(E,D)
Sobrelamesa(B)	Sobre(A, B) ^ Sobre(E, D) ^ Despejado (A) ^	TOMAR(A)
Sobrelamesa(A)	Despejado (E) ^ Despejado (C) ^	APILAR(A,B)
Brazolibre()	Sobrelamesa (B) ^ Sobrelamesa (D) ^	
Sobre(E,D)	Sobrelamesa (C) ^ brazolibre()	
Agarrado(A)		
Brazolibre()		
Sobre(A,B)		

Ejercicio 5

5)



Estado Inicial



Estado Objetivo

Estado Actual		
Sobrelamesa(E)	✓ Sobrelamesa (E)	
Sobrelamesa(B)	✓ Sobrelamesa (C)	
Sobrelamesa(C)	Sobrelamesa (D)	
Despejado(D)	Sobre(A, D)	
Despejado(A)	Sobre(B, E)	
Despejado(C)	Despejado (A)	
Brazolibre()	Despejado (B)	
Sobre(D, E)	Despejado (C)	
Sobre(A, B)	brazolibre()	
	Sobre(A, D) ^ Sobre(B, E) ^ Despejado (A) ^	
	Despejado (B) ^ Despejado (C) ^ Sobrelamesa (D) ^	
	Sobrelamesa (E) ^ Sobrelamesa (C) ^ brazolibre()	
Sobrelamesa(E)	✓ Sobre(D,E)	DESAPILAR (D,E)
Sobrelamesa(B)	✓ despejado (D)	
Sobrelamesa(C)	✓ brazolibre()	
Despejado(D)	✓ Sobre(D,E) ^ despejado (D) ^ brazolibre()	
Despejado(A)	DESAPILAR (D,E)	
Despejado(C)	Sobre(A, D)	
Brazolibre()	Sobre(B, E)	
Sobre(D, E)	Despejado (A)	
	Despejado (B)	

Sobre(A, B) Agarrado(D) Despejado(E)	Despejado (C) brazolibre() Sobre(A, D) ^ Sobre(B, E) ^ Despejado (A) ^ Despejado (B) ^ Despejado (C) ^ Sobrelamesa (D) ^ Sobrelamesa (E) ^ Sobrelamesa (C) ^ brazolibre()	
Sobrelamesa(E) Sobrelamesa(B) Sobrelamesa(C) Despejado(D) Despejado(A) Despejado(C) Sobre(A, B) Agarrado(D) Despejado(E)	✓ Despejado(D) Agarrado(A) Despejado(D) ^ agarrado(A) APILAR(A, D) Sobre(B, E) Despejado (A) Despejado (B) Despejado (C) brazolibre() Sobre(A, D) ^ Sobre(B, E) ^ Despejado (A) ^ Despejado (B) ^ Despejado (C) ^ Sobrelamesa (D) ^ Sobrelamesa (E) ^ Sobrelamesa (C) ^ brazolibre()	DESAPILAR(D,E)
Sobrelamesa(E) Sobrelamesa(B) Sobrelamesa(C) Despejado(D) Despejado(A) Despejado(C) Sobre(A, B) Agarrado(D) Despejado(E)	✓ Despejado(A) Brazolibre() Sobrelamesa(A) Despejado(A) ^ Sobrelamesa(A) ^ Brazolibre() TOMAR(A) Despejado(D) ^ agarrado(A) APILAR(A, D) Sobre(B, E) Despejado (A) Despejado (B) Despejado (C) brazolibre() Sobre(A, D) ^ Sobre(B, E) ^ Despejado (A) ^ Despejado (B) ^ Despejado (C) ^ Sobrelamesa (D) ^ Sobrelamesa (E) ^ Sobrelamesa (C) ^ brazolibre()	
Sobrelamesa(E) Sobrelamesa(B) Sobrelamesa(C) Despejado(D) Despejado(A) Despejado(C) Sobre(A, B) Agarrado(D) Despejado(E)	✓ Agarrado(D) BAJAR(D) Sobrelamesa(A) Agarrado(A) BAJAR(A) Despejado(D) ^ agarrado(A) APILAR(A, D) Sobre(B, E) Despejado (A)	

Brazolibre() Sobrelamesa(D)	Despejado (B) Despejado (C) brazolibre() $\text{Sobre}(A, D) \wedge \text{Sobre}(B, E) \wedge \text{Despejado}(A) \wedge$ $\text{Despejado}(B) \wedge \text{Despejado}(C) \wedge \text{Sobrelamesa}(D) \wedge$ $\text{Sobrelamesa}(E) \wedge \text{Sobrelamesa}(C) \wedge \text{brazolibre}()$	
Sobrelamesa(E) Sobrelamesa(B) Sobrelamesa(C) Despejado(D) Despejado(A) Despejado(C) Sobre(A, B) Despejado(E) Brazolibre() Sobrelamesa(D) Agarrado(A) Despejado(B) Brazolibre() Sobre(A,D)	✓ Sobre(A,B) ✓ despejado(A) ✓ brazolibre() ✓ $\text{Sobre}(A,B) \wedge \text{despejado}(A) \wedge \text{brazolibre}()$ DESAPILAR (A,B) ✓ Agarrado(A) ✓ $\text{Despejado}(D) \wedge \text{agarrado}(A)$ APILAR(A, D) Sobre(B, E) Despejado (A) Despejado (B) Despejado (C) brazolibre() $\text{Sobre}(A, D) \wedge \text{Sobre}(B, E) \wedge \text{Despejado}(A) \wedge$ $\text{Despejado}(B) \wedge \text{Despejado}(C) \wedge \text{Sobrelamesa}(D) \wedge$ $\text{Sobrelamesa}(E) \wedge \text{Sobrelamesa}(C) \wedge \text{brazolibre}()$	
Sobrelamesa(E) Sobrelamesa(B) Sobrelamesa(C) Despejado(A) Despejado(C) Despejado(E) Sobrelamesa(D) Despejado(B) Brazolibre() Sobre(A,D)	✓ Despejado(E) Agarrado(B) $\text{Despejado}(E) \wedge \text{agarrado}(B)$ APILAR(B, E) Despejado (A) Despejado (B) Despejado (C) brazolibre() $\text{Sobre}(A, D) \wedge \text{Sobre}(B, E) \wedge \text{Despejado}(A) \wedge$ $\text{Despejado}(B) \wedge \text{Despejado}(C) \wedge \text{Sobrelamesa}(D) \wedge$ $\text{Sobrelamesa}(E) \wedge \text{Sobrelamesa}(C) \wedge \text{brazolibre}()$	
Sobrelamesa(E) Sobrelamesa(B) Sobrelamesa(C) Despejado(A) Despejado(C) Despejado(E) Sobrelamesa(D) Despejado(B) Brazolibre() Sobre(A,D)	✓ Despejado(B) ✓ Sobrelamesa(B) ✓ Brazolibre() ✓ $\text{Despejado}(B) \wedge \text{Sobrelamesa}(B) \wedge \text{Despejado}(B)$ TOMAR(B) ✓ $\text{Despejado}(E) \wedge \text{agarrado}(B)$ APILAR(B, E) ✓ Despejado (A) ✓ Despejado (B) ✓ Despejado (C) ✓ brazolibre()	

Agarrado(B) Sobre(B,E) Brazolibre()	✓ Sobre(A, D) ^ Sobre(B, E) ^ Despejado (A) ^ Despejado (B) ^ Despejado (C) ^ Sobrelamesa (D) ^ Sobrelamesa (E) ^ Sobrelamesa (C) ^ brazolibre()	
---	---	--

Lógica Difusa

Ejercicio 1

Defina que es un Sistema de Inferencia basado en Lógica Difusa.

Un Sistema de Inferencia basado en Lógica Difusa es un tipo de sistema que utiliza la lógica difusa para procesar información imprecisa o incierta. En lugar de seguir la lógica binaria convencional (verdadero o falso), la lógica difusa permite representar y manipular grados de veracidad en un rango continuo, lo que es especialmente útil cuando se trabaja con conceptos ambiguos o variables con valores no absolutos.

Ejercicio 2

Mencione las etapas en el proceso de resolución en un Sistema de Inferencia basado en Lógica Difusa y explique brevemente cada una.

1. Fuzzificación:

- Evaluamos las entradas en las distintas funciones de membresía.
- Se determina el grado en el que cada entrada pertenece a cada uno de los conjuntos difusos.
- Se evalúan las funciones de membresía.
- La entrada es siempre un valor numérico limitado al universo del discurso de la variable de entrada y la salida es un grado difuso de pertenencia (siempre intervalo entre 0 y 1).

2. Aplicación de operadores difusos

Una vez que las entradas han sido fuzzificadas, conocemos el grado en el cual cada parte del antecedente ha sido satisfecho para cada regla.

- Si alguna regla tiene un antecedente compuesto, se deben aplicar los operadores para obtener un único número que representa el antecedente.
- Cada operador difuso recibe uno o más valores y devuelve un único valor de verdad.

Operador	Método	Expresión
P y Q	Mínimo	$\min(P, Q)$
"	Producto	$P * Q$
"	Truncamiento	$\max((P + Q - 1), 0)$
P o Q	Máximo	$\max(P, Q)$
"	Amplificación	$P + Q * (1 - P)$
"	Adición	$\min(P + Q, 1)$
no P	Complemento	$1 - P$

3. Aplicación del método de implicación

El método de implicación se define como la conformación del consecuente (un conjunto fuzzy) basado en el antecedente (un número). La entrada para la implicación es un número dado por el antecedente, y la salida es un conjunto fuzzy. La se obtiene para cada regla por separado.

Algunos métodos podrían ser mín. (mínimo) que trunca el conjunto fuzzy, y prod. (producto) el que amplifica el conjunto de salida fuzzy.

- Conformar el consecuente (conjunto difuso) para cada regla.
- El resultado es proporcional al valor del antecedente.
- Las reglas se evalúan individualmente para obtener el valor de la implicación. Luego deben asociarse los conjuntos de salida obtenidos para cada regla.

4. Agregación

- Se unifican las salidas uniando los procesos paralelos.
- Se combinan las salidas en un único conjunto difuso.
- La entrada del proceso es la lista de las salidas truncadas según el resultado de la implicación.
- La salida es un único conjunto difuso para cada variable de salida.
- Algunos métodos son: max (máximo), prob-or (or probabilística), y sum (la suma de cada conjunto de salida para cada regla).
- El proceso de agregación consiste en formar un único conjunto de salida para todo el sistema a partir de los conjuntos de salida de cada regla.

5. Defuzzificación

- La entrada para el proceso de defuzzificación es el conjunto difuso agregado (el conjunto difuso unificado de salida) y la salida es un número.
- La defuzzificación tiene lugar en dos distintos pasos. Primero las funciones de pertenencia son escaladas de acuerdo a sus posibles valores, luego estas son usadas para calcular el centroide de los conjuntos difusos asociados.
- El método más popular es el cálculo del centroide, el cual retorna el centro de un área bajo una curva.

$$CG = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{x=a}^b \mu_A(x)}$$

i. Numerador: fragmentos * riesgo

ii. Denominador: cantidad de fragmentos * riesgo

Ejercicio 3

Suponiendo un tanque como el de la fig. 1, el cual recibe un flujo de líquido de acuerdo a la apertura de una válvula V1, la estrategia para la toma de decisión en cuanto al manejo de esta válvula, podría ser descrita por un experto siguiendo una serie de reglas empíricas. Estas reglas se pueden utilizar mediante un sistema de inferencia difuso para manejar la apertura de V1 mediante una señal s , donde un ejemplo de cómo podría implementarse para una situación particular dada, de acuerdo a los valores que toma la presión P en la parte superior del tanque y la temperatura T en el interior del fluido, se muestra mediante los ejercicios que siguen:

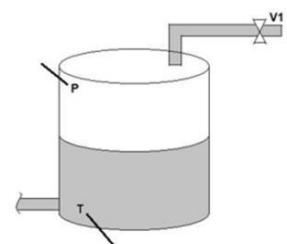
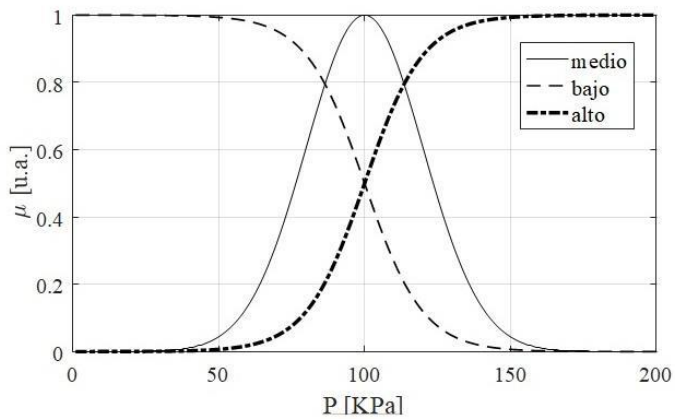


Figura 1|

Ejercicio 3.1

Fuzzificar la variable $P=110\text{KPa}$, cuyo rango va de 0°KPa a 200KPa , según los siguientes conjuntos difusos:



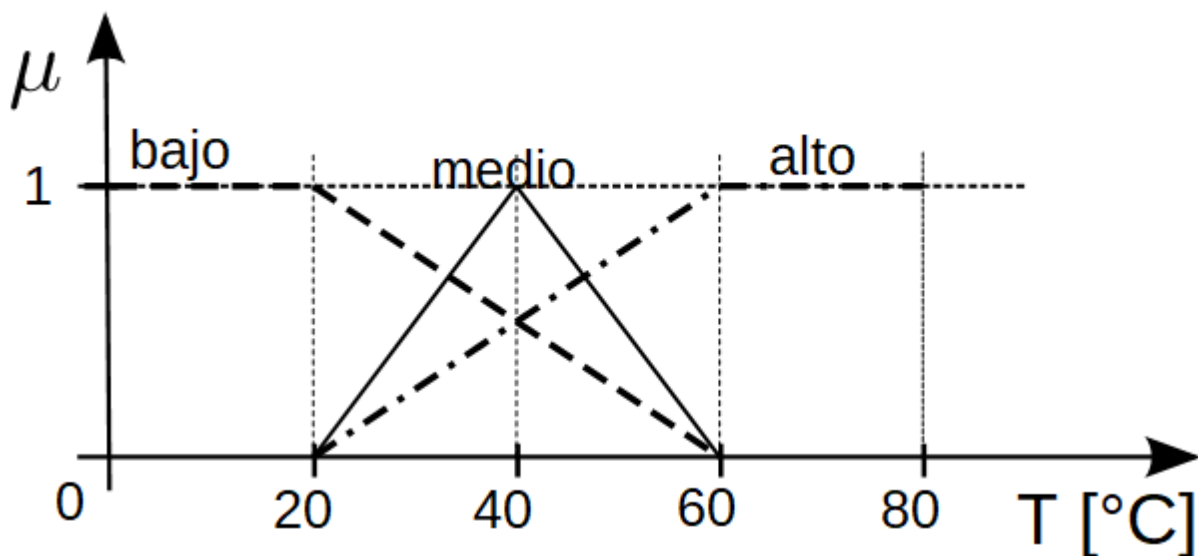
$$\left\{ \mu_{bajo} = 1 - \frac{1}{1 + e^{-0,1(P - 100)}}; \mu_{medio} = e^{-\frac{(P - 100)^2}{800}}; \mu_{alto} = \frac{1}{1 + e^{-0,1(P - 100)}} \right\}$$

Reemplazando $P = 110\text{KPa}$ en (1), entonces se tiene que P_{difuso} es:

- bajo con grado de pertenencia $\mu_{bajo} = 0,269$
- medio con grado de pertenencia $\mu_{medio} = 0,882$
- alto con grado de pertenencia $\mu_{alto} = 0,731$

Ejercicio 3.2

Fuzzificar la variable $T = 35^\circ\text{C}$, cuyo rango va de 0°C a 80°C , según los conjuntos difusos definidos a continuación:



Dato: ecuación de la recta a partir del gráfico:

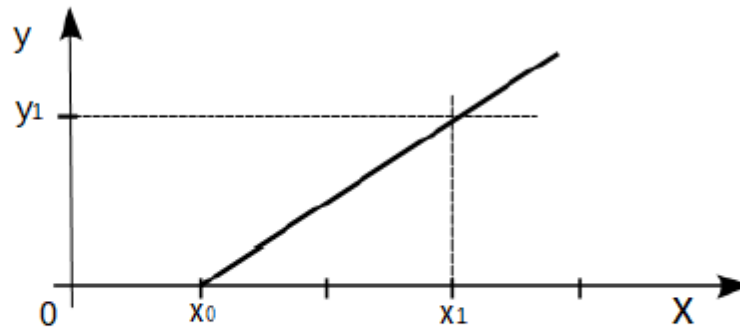


Figura 3

$$y = \frac{y_1}{(x_1 - x_0)}(x - x_0)$$

Definición paramétrica de cada conjunto difuso, utilizando la ecuación de la recta y la representación gráfica de los conjuntos difusos, según se muestra en el enunciado:

$$\mu_{bajo} = \begin{cases} 1 \quad \forall T \leq 20 \\ \frac{1}{20-60}(T - 60) = -\frac{T}{40} + 1,5 \quad \forall 20 < T \leq 60 \\ 0 \quad \forall T > 60 \end{cases}$$

$$\mu_{medio} = \begin{cases} 0 \quad \forall 60 \leq T \leq 20 \\ \frac{1}{40-20}(T - 20) = \frac{T}{20} - 1 \quad \forall 20 < T \leq 40 \\ \frac{1}{40-60}(T - 60) = -\frac{T}{20} + 3 \quad \forall 40 < T < 60 \end{cases}$$

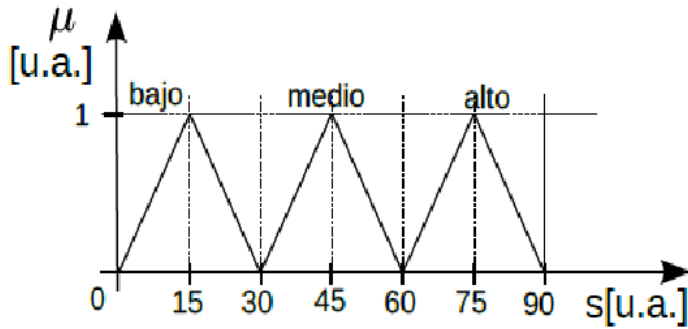
$$\mu_{alto} = \begin{cases} 0 \quad \forall T \leq 20 \\ \frac{1}{60-20}(T - 20) = \frac{T}{40} - 0,5 \quad \forall 20 < T < 60 \\ 1 \quad \forall 60 \leq T \end{cases}$$

Luego, para $T=35^\circ\text{C}$, se tendrá que T_{difuso} es:

- bajo con grado de pertenencia $\mu_{bajo} = 0,625$
- medio con grado de pertenencia $\mu_{medio} = 0,75$
- alto con grado de pertenencia $\mu_{alto} = 0,375$

Ejercicio 3.3

Dado un sistema de inferencia difusa cuyas entradas están dadas por las variables T y P según se definió en los ejercicios 3.1) y 3.2) y cuya salida es una variable difusa normalizada, que aplica a un dominio $s = \{0 \leq R \leq 90\}$, y contiene tres conjuntos difusos: *bajo*, *medio*, *alto*; definidos como sigue:



Se pide encontrar cual será el conjunto difuso que resulte de evaluar cada una de las siguientes reglas de inferencia:

- a) If T es bajo AND P es bajo, THEN s es alto
- b) If T es medio OR P es bajo, THEN s es medio

Considerar el Sistema de Inferencia Difuso y resolver los operadores lógicos de variables difusas.

De acuerdo al sistema de inferencia Mamdani, se aplicará la operación lógica que corresponda entre los conjuntos difusos antecedentes, para obtener el consecuente de cada regla. En el caso de la técnica de Zadeh será la función $\min(\cdot)$ para la operación AND y la función $\max(\cdot)$ para la operación OR. Entonces se tiene:

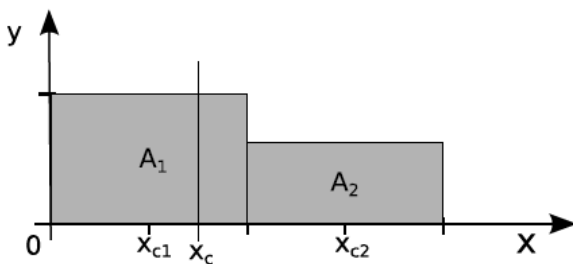
- a) $(T_{\text{bajo}}, \mu_{\text{bajo}}(T) = 0,625) \text{ AND } (P_{\text{bajo}}, \mu_{\text{bajo}}(P) = 0,269) = (\text{Salto}, \min(\mu_{\text{bajo}}(T), \mu_{\text{bajo}}(P))) = (\text{Salto}, \mu_{\text{alto}}(s) = 0,269)$
- b) $(T_{\text{medio}}, \mu_{\text{medio}}(T) = 0,75) \text{ OR } (P_{\text{bajo}}, \mu_{\text{bajo}}(P) = 0,269) = (S_{\text{medio}}, \max(\mu_{\text{medio}}(T), \mu_{\text{bajo}}(P))) = (S_{\text{medio}}, \mu_{\text{medio}}(s) = 0,75)$

Ejercicio 3.4

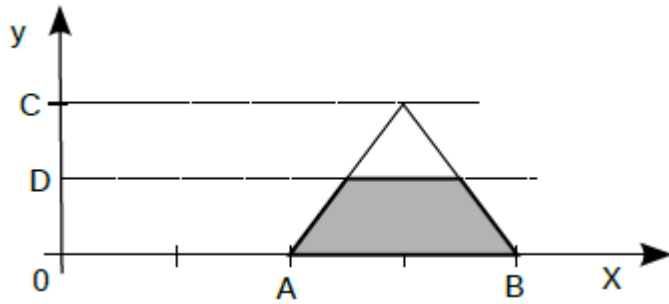
Si a partir de las dos reglas definidas en el ejercicio 3.3) para un sistema de inferencia difuso se quiere obtener un único valor de s , defusificando utilizando el método del centroide o centro de gravedad; cual será entonces la magnitud de s .

Para obtener un único valor difuso para s a partir de los dos valores obtenidos en el ejercicio 3), se aplica el proceso de agregación, donde se truncan los conjuntos de salida de cada regla al valor del grado de pertenencia asociado. Luego se realiza la unión de los conjuntos difusos obtenidos, para finalmente defuzificar el conjunto difuso y obtener un valor Crisp.

Debido a que los conjuntos difusos bajo, alto y medio no están solapados, puede encontrarse el centroide de la superficie que resulta de la unión de dichos conjuntos, como el centroide de una figura compuesta (figura 5). El valor de la componente x de las coordenadas del centroide de la figura compuesta por la suma de A_1 y A_2 (toda el área coloreada), y en general la suma de n áreas que forman el área total de la cual se calcula el centroide será:



Luego, se tienen tres triángulos isósceles correspondientes a bajo, medio y alto, a los que llamaremos T_1 , T_2 y T_3 . Para calcular el centroide entonces, se tiene que calcular antes el área de un triángulo isósceles truncado hasta una altura dada.



Para calcular el área de un triángulo isósceles truncado hasta una altura D (área coloreada figura 6, AT) se hace:

$$A_T = 2 \left[\left(\frac{B-A}{2} - \frac{\frac{B-A}{2}D}{C} \right) D + \frac{(\frac{B-A}{2}D)D}{2} \right]$$

Si $C=1$

$$A_T = 2 \left[\frac{B-A}{2} D - \frac{B-A}{2} D^2 + \frac{B-A}{4} D^2 \right] = 2(B-A)D \left(1 - \frac{D}{2} \right)$$

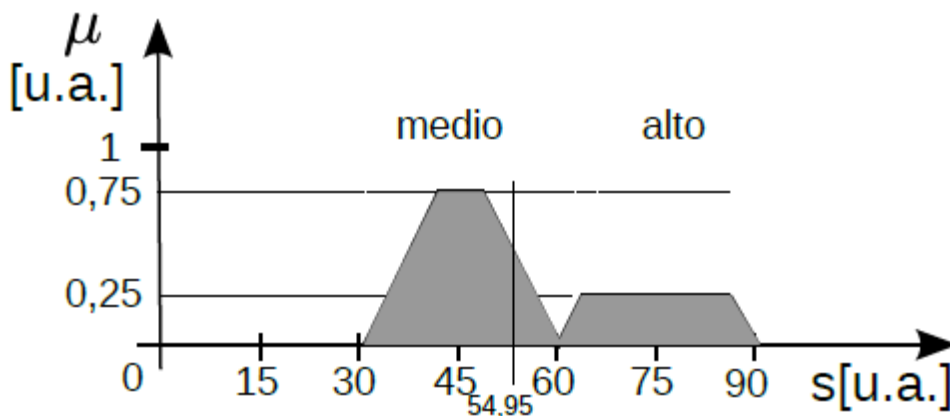
Considerando que la regla a) del ejercicio 3) genera una pertenencia de s al conjunto difuso alto con grado de pertenencia de 0,269 mientras que la regla b) del ejercicio 3) genera una pertenencia de s al conjunto difuso medio con grado de pertenencia de 0,75; entonces, se debe encontrar el área para T_3 truncado a una altura de 0,269 y para T_2 truncado a una altura de 0,75.

$$A_{T_3} = (90 - 60) * 0.269 \left(\frac{1 - 0.269}{2} \right) = 6.98$$

$$A_{T_2} = (60 - 30) * 0.75 \left(\frac{1 - 0.75}{2} \right) = 14.06$$

Como el triángulo truncado de la figura 6 es simétrico respecto de un eje paralelo al eje y, la coordenada x de su centroide estará por donde pase este eje de simetría. Finalmente, la salida Crisp del sistema será:

$$S_{Crisp} = \frac{x_{c2}A_{T_2} + x_{c3}A_{T_3}}{A_{T_2} + A_{T_3}} = \frac{45 * 14.06 + 75 * 6.98}{14.06 + 6.98} = 54.95$$



Reconocimiento de Patrones

Ejercicio 1

Defina: reconocimiento de patrones, patrón, clase.

Reconocimiento de Patrones: El reconocimiento de patrones es una rama de la inteligencia artificial que se centra en identificar y clasificar patrones en datos o información. Se refiere a la capacidad de un sistema para aprender y reconocer relaciones, regularidades o características distintivas en un conjunto de datos para tomar decisiones o realizar predicciones.

Patrón: Un patrón es una característica, estructura o conjunto de datos que se repite o se encuentra en múltiples instancias dentro de un conjunto de datos. Los patrones pueden ser visuales, numéricos, textuales o de otro tipo, y son identificables debido a su consistencia o similitud en varias observaciones.

Clase: En el contexto del reconocimiento de patrones, una clase es una categoría o etiqueta a la que se asigna un conjunto de datos o un patrón. Las clases se utilizan para agrupar y organizar los datos en categorías significativas, lo que permite al sistema de reconocimiento de patrones clasificar nuevos patrones en una de estas categorías. Por ejemplo, en el reconocimiento de caracteres escritos a mano, las clases podrían representar letras del alfabeto o dígitos numéricos.

Ejercicio 2

Dados los siguientes vectores de entrada (0 0) (0 1) (1 0) (1 1) determinar la función de decisión lineal que realiza la correcta clasificación de las mismas según el siguiente vector de salidas $y = [1 \ 1 \ -1 \ -1]$.

Dato: el cálculo de la inversa es

$$(X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad X^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$\begin{aligned} A &= (X^T X)^{-1} * X^T * Y = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} -0.5 & -0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \\ 0.75 & 0.25 & 0.25 & -0.25 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = [-2 \quad 0 \quad 1] \end{aligned}$$

Ejercicio 3

Dados los siguientes vectores de entrada (0 0) (1 0) (0 1) (1 1) determinar la función de decisión lineal que realiza la correcta clasificación de las mismas según el siguiente vector de salidas $y = [1 \ 1 \ -1 \ -1]$.

Dato: el cálculo de la inversa es

$$(X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad X^T = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$\begin{aligned} A &= (X^T X)^{-1} * X^T * Y = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} -0.5 & 0.5 & -0.5 & 0.5 \\ -0.5 & -0.5 & 0.5 & 0.5 \\ 0.75 & 0.25 & 0.25 & -0.25 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = [0 \quad -2 \quad 1] \end{aligned}$$

Ejercicio 4

Dados los siguientes vectores de entrada (0 0) (0 1) (1 0) (1 1) determinar la función de decisión lineal que realiza la correcta clasificación de las mismas según el siguiente vector de salidas $y = [1 \ -1 \ -1 \ -1]$. En este caso se trata de la función lógica OR.

Dato: el cálculo de la inversa es

$$(X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad X^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$\begin{aligned} A &= (X^T X)^{-1} * X^T * Y = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} -0.5 & -0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \\ 0.75 & 0.25 & 0.25 & -0.25 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = [-1 \quad -1 \quad 0.5] \end{aligned}$$

Ejercicio 5

Dados los siguientes vectores de entrada (0 0) (0 1) (1 0) (1 1) determinar la función de decisión lineal que realiza la correcta clasificación de las mismas según el siguiente vector de salidas $y = [1 \ 1 \ 1 \ -1]$ En este caso se trata de la función lógica AND.

Dato: el cálculo de la inversa es

$$(X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad X^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (X^T X)^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix}$$

$$\begin{aligned} A &= (X^T X)^{-1} * X^T * Y = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ -0.5 & -0.5 & 0.75 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} -0.5 & -0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \\ 0.75 & 0.25 & 0.25 & -0.25 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = [-1 \quad -1 \quad 1.5] \end{aligned}$$

Ejercicio 6

Dados los siguientes vectores de entrada (0 0) (0 1) (1 0) (2 2) determinar la función de decisión lineal que realiza la correcta clasificación de las mismas según el siguiente vector de salidas $y = [1 \ 1 \ 1 \ -1]$.

Dato: el cálculo de la inversa es

$$(X^T X)^{-1} = \begin{bmatrix} 0.6 & -0.38 & -0.16 \\ -0.38 & 0.61 & -0.16 \\ -0.16 & -0.16 & 0.5 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 2 & 2 & 1 \end{bmatrix} \quad X^T = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (X^T X)^{-1} = \begin{bmatrix} 0.6 & -0.38 & -0.16 \\ -0.38 & 0.61 & -0.16 \\ -0.16 & -0.16 & 0.5 \end{bmatrix}$$

$$A = (X^T X)^{-1} * X^T * Y = \begin{bmatrix} 0.6 & -0.38 & -0.16 \\ -0.38 & 0.61 & -0.16 \\ -0.16 & -0.16 & 0.5 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.16 & -0.54 & 0.44 & 0.28 \\ -0.16 & 0.45 & -0.54 & 0.3 \\ 0.5 & 0.34 & 0.34 & -0.14 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = [-0.54 \quad -0.55 \quad 1.32]$$

Ejercicio 7

Dados los siguientes vectores de entrada (0 0) (1 0) (2 0) (0 1) (1 1) (2 1) determinar la función de decisión lineal que realiza la correcta clasificación de las mismas según el siguiente vector de salidas: $y = [1 \ 1 \ 1 \ -1 \ -1 \ -1]$

Dato: el cálculo de la inversa es

$$(X^T X)^{-1} = \begin{bmatrix} 0.25 & 0 & -0.25 \\ 0 & 0.66 & -0.33 \\ -0.25 & -0.33 & 0.583 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \quad X^T = \begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (X^T X)^{-1} = \begin{bmatrix} 0.25 & 0 & -0.25 \\ 0 & 0.66 & -0.33 \\ -0.25 & -0.33 & 0.583 \end{bmatrix}$$

$$A = (X^T X)^{-1} * X^T * Y = \begin{bmatrix} 0.25 & 0 & -0.25 \\ 0 & 0.66 & -0.33 \\ -0.25 & -0.33 & 0.583 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.25 & 0 & 0.25 & -0.25 & 0 & 0.25 \\ -0.33 & -0.33 & -0.33 & 0.33 & 0.33 & 0.33 \\ 0.583 & 0.333 & 0.083 & 0.253 & 0.003 & -0.247 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = [0 \quad -1.98 \quad 0.99]$$

Redes Neuronales

Ejercicio 1

Defina: redes neurales. Topología de una red neuronal. Aprendizaje de una red neuronal.

Redes Neuronales: Las redes neuronales son modelos computacionales inspirados en el funcionamiento del cerebro humano. Están compuestas por un conjunto de nodos (neuronas artificiales) interconectados que trabajan en paralelo para procesar información. Estas redes se utilizan para tareas de aprendizaje automático y procesamiento de datos, especialmente en reconocimiento de patrones, clasificación y regresión.

Topología de una Red Neuronal: La topología de una red neuronal se refiere a la arquitectura o disposición de sus neuronas y conexiones. Esto incluye el número de capas, el número de neuronas en cada capa y cómo están conectadas entre sí. Ejemplos de topologías comunes son las redes neuronales feedforward, las redes neuronales recurrentes y las redes neuronales convolucionales.

Aprendizaje de una Red Neuronal: El aprendizaje en una red neuronal se refiere al proceso de ajustar los pesos de las conexiones entre las neuronas para que la red pueda realizar una tarea específica. Hay dos tipos principales de aprendizaje en redes neuronales:

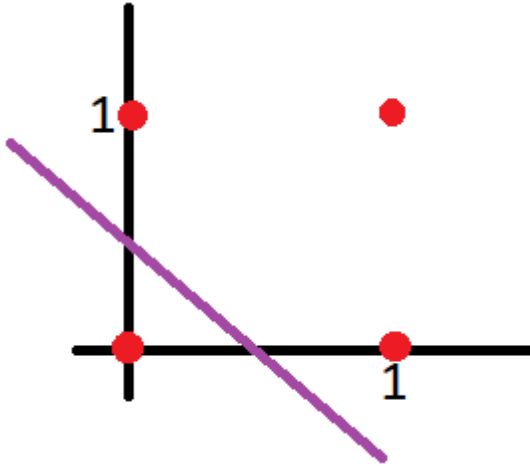
- **Aprendizaje Supervisado:** La red se entrena con un conjunto de datos que incluye entradas y las respuestas deseadas (etiquetas). La red ajusta sus pesos para minimizar la diferencia entre las respuestas reales y las predicciones.
- **Aprendizaje No Supervisado:** La red se entrena en datos sin etiquetas y busca aprender patrones o estructuras ocultas en los datos, como agrupaciones o reducción de dimensionalidad.

En ambos casos, el aprendizaje implica la actualización iterativa de los pesos de las conexiones mediante algoritmos como retropropagación (backpropagation) en el aprendizaje supervisado.

Ejercicio 2

Dada una Red perceptron que reproduce el comportamiento de una compuerta lógica OR. Se pide:

- a) Sabiendo que las salidas posibles para dicha función son 0 y 1, graficar la solución a la que se arribará.



- b) Dado el siguiente cuadro de iteración de la red perceptron para la función lógica OR responder:

- i. ¿Qué representan los tres elementos que se dan como entrada?
El umbral (siempre tiene valor 1 y se lo considera una entrada más) y las otras son las entradas de la compuerta OR
- ii. ¿Por qué se modifican los pesos en algunos casos?
Los pesos se modifican porque hubo error (la salida deseada y la calculada no son iguales)
- iii. ¿Cómo se obtiene la columna de la "Salida Calculada"?
Se aplica la función $\sum X_n * W_n$ (Sumatoria del producto de cada entrada por su peso)

(x,s)		Pesos	Salida	Ajuste de pesos	Error
Entrada	Salida				
1 0 0	0	-2,5 0,5 1,5	$f(-2,5) = 0$	$\begin{Bmatrix} -2,5 \\ 0,5 \\ 1,5 \end{Bmatrix} + (0) \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -2,5 \\ 0,5 \\ 1,5 \end{Bmatrix}$	0
1 0 1	1	-2,5 0,5 1,5	$f(-1,0) = 0$	$\begin{Bmatrix} -2,5 \\ 0,5 \\ 1,5 \end{Bmatrix} + (1) \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -1,5 \\ 0,5 \\ 2,5 \end{Bmatrix}$	1
1 1 0	1	-1,5 0,5 2,5	$f(-1,0) = 0$	$\begin{Bmatrix} -1,5 \\ 0,5 \\ 2,5 \end{Bmatrix} + (1) \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -0,5 \\ 1,5 \\ 2,5 \end{Bmatrix}$	1
1 1 1	1	-0,5 1,5 2,5	$f(3,5) = 1$	$\begin{Bmatrix} -0,5 \\ 1,5 \\ 2,5 \end{Bmatrix} + (0) \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -0,5 \\ 1,5 \\ 2,5 \end{Bmatrix}$	0

Ejercicio 3

Dado un perceptron en una red neural de una capa, con dos entradas y una salida, ajuste los pesos asociados a cada entrada para que la salida de este sistema responda según la función OR de las entradas. La función de activación estará dada por:

$$y = \begin{cases} 1 & \text{si } u > 0 \\ 0 & \text{si } u \leq 0 \end{cases} \quad \text{Siendo:} \quad u = \sum w_j \cdot x_j = w_1 * x_1 + w_2 * x_2 + w_3$$

Con los pesos:

a) $w_0 = [1,5 \ 0,5 \ 1,5]$ -> Condiciones iniciales del vector de pesos. Se representan las entradas de la función con x_1 y x_2 . Factor $\alpha = 1$. ($w_1 = 1,5 \ w_2 = 0,5 \ w_3 = 1,5$)

b) $w_0 = [1,5 \ 1 \ 1,5]$ -> Condiciones iniciales del vector de pesos. Factor $\alpha = 1$.

w_j es el vector de pesos. x_j son las entradas.

a)

Entradas	Salida Esperada	Pesos	Salida Obtenida	Error	Ajuste
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 1,5 \\ 1,5 \\ 0,5 \end{bmatrix}$	$u = 1 * 1,5 + 0 * 1,5 + 0 * 0,5 = 1,5$ $y(1,5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 1,5 \\ 1,5 \\ 0,5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,5 \\ 1,5 \\ 0,5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0,5 \\ 1,5 \\ 0,5 \end{bmatrix}$	$u = 1 * 0,5 + 0 * 1,5 + 1 * 0,5 = 1$ $y(1) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste

$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 0 * 0.5 = 2$ $y(2) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 1 * 0.5 = 2.5$ $y(2.5) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste
Segunda iteración					
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 0 * 1.5 + 0 * 0.5 = 0.5$ $y(0.5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} -0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * -0.5 + 0 * 1.5 + 1 * 0.5 = 0$ $y(0) = 0$	$e = 1 - 0$ $e = 1$	$\begin{bmatrix} -0.5 \\ 1.5 \\ 0.5 \end{bmatrix} + 1 * (1) * \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1.5 \\ 1.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 1.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 0 * 1.5 = 2$ $y(2) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 1.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 1 * 1.5 = 2.5$ $y(2.5) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste

b)

Entradas	Salida Esperada	Pesos	Salida Obtenida	Error	Ajuste
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 1.5 \\ 1.5 \\ 1 \end{bmatrix}$	$u = 1 * 1.5 + 0 * 1.5 + 0 * 1 = 1.5$ $y(1.5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 1.5 \\ 1.5 \\ 1 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1.5 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 0 * 1.5 + 1 * 0.5 = 1$ $y(1) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 0 * 0.5 = 2$ $y(2) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 1 * 0.5 = 2.5$ $y(2.5) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste
Segunda iteración					
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 0 * 1.5 + 0 * 0.5 = 0.5$ $y(0.5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} -0.5 \\ 1.5 \\ 0.5 \end{bmatrix}$	$u = 1 * -0.5 + 0 * 1.5 + 1 * 0.5 = 0$ $y(0) = 0$	$e = 1 - 0$ $e = 1$	$\begin{bmatrix} -0.5 \\ 1.5 \\ 0.5 \end{bmatrix} + 1 * (1) * \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1.5 \\ 1.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 1.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 0 * 1.5 = 2$ $y(2) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste

$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 1.5 \\ 1.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 1.5 + 1 * 1.5 = 2.5$ $y(2.5) = 1$	$e = 1 - 1$ $e = 0$	No hay ajuste
---	---	---	---	------------------------	---------------

Ejercicio 4

Dado un perceptron en una red neural de una capa, con dos entradas y una salida, ajuste los pesos asociados a cada entrada para que la salida de este sistema responda según la función AND de las entradas. La función de activación estará dada por:

$$y = \begin{cases} 1 & \text{si } u > 0 \\ 0 & \text{si } u \leq 0 \end{cases} \quad \text{Siendo:} \quad u = \sum w_j \cdot x_j = w_1 * x_1 + w_2 * x_2 + w_3$$

Con los pesos:

a) $w_0 = [0 \ -2.5 \ 1.5]$ ($w_1 = 0 \ w_2 = -2.5 \ w_3 = 1.5$)

Factor $\alpha = 1$

b) $w_0 = [1.5 \ 0.5 \ 2.5]$ Factor $\alpha = 1$

Ítem a)

Entradas	Salida Esperada	Pesos	Salida Obtenida	Error	Ajuste
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 1.5 \\ 0 \\ -2.5 \end{bmatrix}$	$u = 1 * 1.5 + 0 * 0 + 0 * -2.5 = 1.5$ $y(1.5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 1.5 \\ 0 \\ -2.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ -2.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	0	$\begin{bmatrix} 0.5 \\ 0 \\ -2.5 \end{bmatrix}$	$u = 1 * 0.5 + 0 * 0 + 1 * -2.5 = -2$ $y(-2) = 0$	$e = 0 - 0$ $e = 0$	No hay ajuste
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 0.5 \\ 0 \\ -2.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 0 + 0 * -2.5 = 0.5$ $y(0.5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 0.5 \\ 0 \\ -2.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1 \\ -2.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} -0.5 \\ -1 \\ -2.5 \end{bmatrix}$	$u = 1 * -0.5 + 1 * -1 + 1 * -2.5 = -4$ $y(-4) = 0$	$e = 1 - 0$ $e = 1$	$\begin{bmatrix} -0.5 \\ -1 \\ -2.5 \end{bmatrix} + 1 * (1) * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ -1.5 \end{bmatrix}$
Segunda iteración					
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 0.5 \\ 0 \\ -1.5 \end{bmatrix}$	$u = 1 * 0.5 + 0 * 0 + 0 * -1.5 = -1$ $y(-1) = 0$	$e = 0 - 0$ $e = 0$	$\begin{bmatrix} 0.5 \\ 0 \\ -1.5 \end{bmatrix} + 1 * (0) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ -1.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 0 \\ -1.5 \end{bmatrix}$	$u = 1 * 0.5 + 0 * 0 + 1 * -1.5 = -1$ $y(-1) = 0$	$e = 1 - 0$ $e = 1$	$\begin{bmatrix} 0.5 \\ 0 \\ -1.5 \end{bmatrix} + 1 * (1) * \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0 \\ -0.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	1	$\begin{bmatrix} 1.5 \\ 0 \\ -0.5 \end{bmatrix}$	$u = 1 * 1.5 + 1 * 0 + 0 * -0.5 = 1.5$ $y(1.5) = 1$	$e = 1 - 1$ $e = 0$	$\begin{bmatrix} 1.5 \\ 0 \\ -0.5 \end{bmatrix} + 1 * (0) * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0 \\ -0.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 1.5 \\ 0 \\ -0.5 \end{bmatrix}$	$u = 1 * 1.5 + 1 * 0 + 1 * -0.5 = 1$ $y(1) = 1$	$e = 1 - 1$ $e = 0$	$\begin{bmatrix} 1.5 \\ 0 \\ -0.5 \end{bmatrix} + 1 * (0) * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0 \\ -0.5 \end{bmatrix}$

Ítem b)

Entradas	Salida Esperada	Pesos	Salida Obtenida	Error	Ajuste
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 2.5 \\ 0.5 \\ 1.5 \end{bmatrix}$	$u = 1 * 2.5 + 0 * 0.5 + 0 * 1.5 = 2.5$ $y(2.5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 2.5 \\ 0.5 \\ 1.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.5 \\ 1.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	0	$\begin{bmatrix} 1.5 \\ 0.5 \\ 1.5 \end{bmatrix}$	$u = 1 * 1.5 + 0 * 0.5 + 1 * 1.5 = 3$ $y(3) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 1.5 \\ 0.5 \\ 1.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 0.5 + 0 * 0.5 = 1$ $y(1) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix}$	$u = 1 * -0.5 + 1 * -0.5 + 1 * 0.5 = -0.5$ $y(-0.5) = 0$	$e = 1 - 0$ $e = 1$	$\begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} + 1 * (1) * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 1.5 \end{bmatrix}$
Segunda iteración					
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	0	$\begin{bmatrix} 0.5 \\ 0.5 \\ 1.5 \end{bmatrix}$	$u = 1 * 0.5 + 0 * 0.5 + 0 * 1.5 = 0.5$ $y(0.5) = 1$	$e = 0 - 1$ $e = -1$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 1.5 \end{bmatrix} + 1 * (-1) * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.5 \\ 1.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} -0.5 \\ 0.5 \\ 1.5 \end{bmatrix}$	$u = 1 * -0.5 + 0 * 0.5 + 1 * 1.5 = 1$ $y(1) = 1$	$e = 1 - 1$ $e = 0$	$\begin{bmatrix} -0.5 \\ 0.5 \\ 1.5 \end{bmatrix} + 1 * (0) * \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.5 \\ 1.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$	1	$\begin{bmatrix} -0.5 \\ 0.5 \\ 1.5 \end{bmatrix}$	$u = 1 * -0.5 + 1 * 0.5 + 0 * 1.5 = 0$ $y(0) = 0$	$e = 1 - 0$ $e = 1$	$\begin{bmatrix} -0.5 \\ 0.5 \\ 1.5 \end{bmatrix} + 1 * (1) * \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 2.5 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1	$\begin{bmatrix} 0.5 \\ 0.5 \\ 2.5 \end{bmatrix}$	$u = 1 * 0.5 + 1 * 0.5 + 1 * 2.5 = 3.5$ $y(3.5) = 1$	$e = 1 - 1$ $e = 0$	$\begin{bmatrix} 0.5 \\ 0.5 \\ 2.5 \end{bmatrix} + 1 * (0) * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 2.5 \end{bmatrix}$

Ejercicio 5 – Red Adaline

Dados los siguientes dos patrones de entrada con salida -1 y 1 respectivamente desarrollar los pasos del algoritmo a fin de determinar si el error cuadrático medio es menor a 0.4.

0 0 x	x 0 0
0 x 0	0 x 0
x 0 0	0 0 x
V	V
0 0 1	1 0 0
0 1 0	0 1 0
1 0 0	0 0 1
V	V
Salida = -1	Salida = 1

Asignar estos valores aleatorios a los pesos:

$w_1 = 0.5$
 $w_2 = 0.5$
 $w_3 = -1$
 $w_4 = 1.5$
 $w_5 = 0.5$
 $w_6 = 1.5$
 $w_7 = -1$
 $w_8 = 1.5$
 $w_9 = 0.5$

Entrada	Salida Esp	Pesos	Salida Obtenida	Error	Ajuste
0 0 1	-1	0.5 0.5 -1	$1 * (-1) + 1 * (0.5) + 1 * (-1) = -1.5$	$E = -1 - (-1.5) = 0.5$	$(W_1 + 1) = 0.5 + 1 * 0.5 * 0 = 0.5$

0 1 0 1 0 0		1.5 0.5 1.5 -1 1.5 0.5			$(W2+1) = 0.5 + 1*0.5*0 = 0.5$ $(W3+1) = -1 + 1*0.5*1 = -0.5$ $(W4+1) = 1.5 + 1*0.5*0 = 1.5$ $(W5+1) = 0.5 + 1*0.5*1 = 1$ $(W6+1) = 1.5 + 1*0.5*0 = 1.5$ $(W7+1) = -1 + 1*0.5*1 = -0.5$ $(W8+1) = 1.5 + 1*0.5*0 = 1.5$ $(W9+1) = 0.5 + 1*0.5*0 = 0.5$
1 0 0 0 1 0 0 0 1	1	0.5 0.5 -0.5 1.5 1 1.5 0.5 1.5 0.5	$1*0.5 + 1*1 + 1*0.5 = 2$	$E = 1 - 2 = -1$	$(W1+1) = 0.5 + 1 * (-1) * 1 = -0.5$ $(W2+1) = 0.5 + 1 * (-1) * 0 = 0.5$ $(W3+1) = -0.5 + 1 * (-1) * 0 = -0.5$ $(W4+1) = 1.5 + 1 * (-1) * 0 = 1.5$ $(W5+1) = 1 + 1 * (-1) * 1 = 0$ $(W6+1) = 1.5 + 1 * (-1) * 0 = 1.5$ $(W7+1) = 0.5 + 1 * (-1) * 0 = 0.5$ $(W8+1) = 1.5 + 1 * (-1) * 0 = 1.5$ $(W9+1) = 0.5 + 1 * (-1) * 1 = -0.5$
Error cuadrático medio: $\langle \varepsilon^2 \rangle = \frac{1}{2L} * (0.5^2 + (-1)^2) = \frac{1}{2 * 2} * 1.25 = 0.3125$ Red Estable					

Ejercicio 6 – Red Adaline

Dados los siguientes dos patrones de entrada con salida -1 y 1 respectivamente desarrollar los pasos del algoritmo a fin de determinar si el error cuadrático medio es menor a 0.4.

x 0 0	0 0 x
0 x 0	0 x 0
0 0 x	x 0 0
V	V
1 0 0	0 0 1
0 1 0	0 1 0
0 0 1	1 0 0
V	V
Salida = 1	Salida = -1

Asignar estos valores aleatorios a los pesos:

$w1 = 2$
 $w2 = -1$
 $w3 = -1.5$
 $w4 = 1$
 $w5 = -0.5$
 $w6 = 0.5$
 $w7 = 1.5$
 $w8 = 0.5$
 $w9 = 0.5$

Entrada	Salida Esp	Pesos	Salida Obtenida	Error	Ajuste
1 0 0 0 1 0 0 0 1	1	2 -1 -1.5 1 -0.5 0.5 1.5 0.5 0.5	$1*(2) + 1*(-0.5) + 1*(0.5) = 2$	$E = 1 - 2 = -1$	$(W1+1) = 2 + 1 * (-1) * 1 = 1$ $(W2+1) = -1 + 1 * (-1) * 0 = -1$ $(W3+1) = -1.5 + 1 * (-1) * 0 = -1.5$ $(W4+1) = 1 + 1 * (-1) * 0 = 1$ $(W5+1) = -0.5 + 1 * (-1) * 1 = -1.5$ $(W6+1) = 0.5 + 1 * (-1) * 0 = 0.5$ $(W7+1) = 1.5 + 1 * (-1) * 0 = 1.5$

					$(W8+1) = 0.5 + 1 * (-1) * 0 = 0.5$ $(W9+1) = 0.5 + 1 * (-1) * 1 = -0.5$
0 0 1 0 1 0 1 0 0	-1	1 -1 -1.5 1 -1.5 0.5 1.5 0.5 -0.5	$1*(-1.5) + 1*(-1.5) + 1*(1.5) = -1.5$	$E = -1 - (-1.5) = 0.5$	$(W1+1) = 1 + 1 * (0.5) * 0 = 1$ $(W2+1) = -1 + 1 * (0.5) * 0 = -1$ $(W3+1) = -1.5 + 1 * (0.5) * 1 = -1$ $(W4+1) = 1 + 1 * (0.5) * 0 = 1$ $(W5+1) = -1.5 + 1 * (0.5) * 1 = -1$ $(W6+1) = 0.5 + 1 * (0.5) * 0 = 0.5$ $(W7+1) = 1.5 + 1 * (0.5) * 1 = 2$ $(W8+1) = 0.5 + 1 * (0.5) * 0 = 0.5$ $(W9+1) = -0.5 + 1*(0.5) * 0 = -0.5$
Error cuadrático medio: $\langle \varepsilon^2 \rangle = \frac{1}{2L} * ((-1)^2 + (0.5)^2) = \frac{1}{2 * 2} * 1.25 = 0.3125$ Red Estable					

Metaheurística – Algoritmos Genéticos

Ejercicio 1

Defina: Metaheurística. Algoritmo genético. Uso de los algoritmos genéticos. Aplicaciones.

Metaheurística: Una metaheurística es un enfoque de optimización general que se utiliza para encontrar soluciones de alta calidad a problemas complejos. A diferencia de los métodos de optimización exactos, las metaheurísticas no garantizan encontrar la solución óptima, pero son efectivas para buscar soluciones subóptimas en un tiempo razonable. Estas técnicas son especialmente útiles cuando se trata de problemas que involucran un espacio de búsqueda grande o cuando no se dispone de información completa sobre el problema.

Algoritmo Genético: Un algoritmo genético es una metaheurística inspirada en la evolución biológica. Está diseñado para encontrar soluciones aproximadas a problemas de optimización y búsqueda. Un algoritmo genético trabaja con una población de soluciones candidatas, que se representan como "individuos" en un espacio de búsqueda. Estos individuos evolucionan a lo largo de generaciones, donde se aplican operadores genéticos como la selección, la recombinación (cruce) y la mutación para producir nuevas soluciones y mejorar la calidad de la población.

Uso de los Algoritmos Genéticos: Los algoritmos genéticos se utilizan en una amplia variedad de aplicaciones, particularmente en problemas de optimización combinatoria y de búsqueda. Algunos de los usos más comunes son:

- Diseño de Ingeniería: Se aplican en la optimización de diseños, como la planificación de redes de comunicación, la disposición de circuitos electrónicos, el diseño de vehículos y la planificación de rutas.
- Programación y Scheduling: Se emplean para la programación de tareas, asignación de recursos y planificación de horarios en campos como la logística, la gestión de proyectos y la manufactura.
- Aprendizaje de Máquinas: Los algoritmos genéticos se utilizan en la selección de características, optimización de hiperparámetros y generación de arquitecturas de redes neuronales en el campo del aprendizaje automático.
- Optimización Numérica: Resuelven problemas de optimización en matemáticas, como la minimización de funciones no lineales.
- Evolución de Estrategias y Juegos: Se aplican en la evolución de estrategias en juegos y simulaciones, así como en la generación automática de estrategias en algoritmos de inteligencia artificial.

Ejercicio 2

Dada la $f(x) = x^2$ con x entero entre 0 y 31, utilice Algoritmo Genético para maximizarla. (Realice tres iteraciones).

Población Inicial: 01101 11000 01000 10011

Primera Iteración:

Cromosoma	X	F(x)	Probabilidad	Copias
01101	13	169	0.14	1
11000	24	576	0.49	2
01000	8	64	0.25	0
10011	19	361	0.31	1
		1170		

Padre 1: 11000

Padre 2: 01101

$\begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{matrix}$

Hijo 1: 11001

Hijo 2: 01100

Padre 3: 11000

Padre 4: 10011

$\begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{matrix}$

Hijo 1: 11111

Hijo 2: 10100

Segunda Iteración:

Cromosoma	X	F(x)	Probabilidad	Copias
11001	25	625	0.36	1
01100	12	144	0.08	0
11011	27	729	0.42	2
10000	16	256	0.15	1
		1754		

Padre 1: 11011

Padre 2: 11001

$\begin{matrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{matrix}$

Hijo 1: 11101

Hijo 2: 11011

Padre 1: 11011

Padre 2: 10000

$$\begin{matrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{matrix}$$

Hijo 3: 11001

Hijo 4: 10011

Segunda Iteración:

Cromosoma	X	F(x)	Probabilidad	Copias
11101	29	841	0.33	2
11011	27	729	0.29	1
11001	25	625	0.24	1
10011	19	361	0.14	0
		2556		

Padre 1: 11101

Padre 2: 11011

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{matrix}$$

Hijo 1: 11011

Hijo 2: 11001

Padre 1: 11101

Padre 2: 11001

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{matrix}$$

Hijo 3: 11001

Hijo 4: 11101

Ejercicio 3

Dada la siguiente función, maximícela empleando Algoritmo Genético usando el lenguaje de programación Python: $f(x) = -(0.1 + (1 - x)^2 - 0.1 * \cos(6 * \pi * (1 - x))) + 2$

```

import numpy as np

# Función de aptitud (fitness function)
def fitness(x):
    return -(0.1 + (1 - x) ** 2 - 0.1 * np.cos(6 * np.pi * (1 - x))) + 2

# Parámetros del algoritmo genético
population_size = 100
generations = 100
mutation_rate = 0.01

# Inicialización de la población
population = np.random.rand(population_size)

for generation in range(generations):
    # Evaluar la aptitud de la población
    fitness_values = np.array([fitness(x) for x in population])

    # Selección basada en la aptitud (selección por ruleta)

```

```

selection_probabilities = fitness_values / fitness_values.sum()

selected_indices = np.random.choice(range(population_size), size=population_size,
p=selection_probabilities)

# Crear una nueva población a partir de la selección
new_population = population[selected_indices]

# Cruzamiento (usando un punto de cruce)
crossover_point = np.random.randint(1, population_size)
for i in range(0, population_size, 2):
    parent1 = new_population[i]
    parent2 = new_population[i + 1]
    new_population[i] = np.concatenate((parent1[:crossover_point], parent2[crossover_point:]))
    new_population[i + 1] = np.concatenate((parent2[:crossover_point], parent1[crossover_point:]))

# Mutación
for i in range(population_size):
    if np.random.rand() < mutation_rate:
        mutation_point = np.random.randint(0, population_size)
        new_population[i][mutation_point] = np.random.rand()

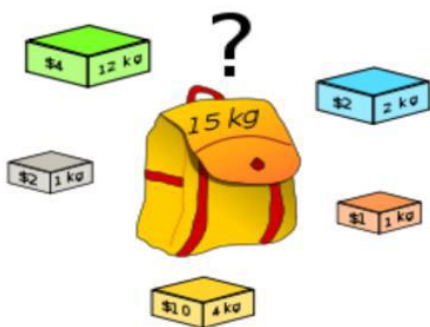
population = new_population

# Encontrar el mejor individuo y su aptitud
best_individual = population[np.argmax(fitness_values)]
best_fitness = max(fitness_values)
print("Mejor individuo:", best_individual)
print("Mejor aptitud:", best_fitness)

```

Ejercicio 4

Dada una mochila con una capacidad de 15 kg que se puede llenar con cajas de distinto peso y valor. ¿Qué cajas elije de modo de maximizar las ganancias y no exceder el peso permitido?



$$Z = 4x_1 + 5x_2 + 6x_3 + 3x_4$$

$$P \rightarrow 7x_1 + 6x_2 + 8x_3 + 2x_4 \leq 15$$

Probabilidad de cruce: 0.98

Probabilidad de mutación: 0.1

Población Inicial: 0110 1010 0100 0110

Cromosoma	Z	P	Prob	ProbAc
0110	11	14	0.2972	0.2972
1010	10	15	0.2702	0.5674
0100	5	6	0.1351	0.7025
0110	11	14	0.2972	0.9997

Probabilidad de primer Padre: 0.43 → Padre 1: 1010

Probabilidad de segundo Padre: 0.69 → Padre 2: 0100

Probabilidad de cruce: 0.84 → $0.84 \leq 0.98$ → Cruzan

Probabilidad de corte: 0.44 → Corta entre el gen 2 y el gen 3

Probabilidades de mutaciones: $\begin{matrix} 0.29 & 0.37 & 0.03 & 0.62 \\ 0.05 & 0.63 & 0.41 & 0.17 \end{matrix}$ → Mutan el gen 3 del padre 1 y el gen 1 del padre 2.

$\begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{matrix} \rightarrow \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{matrix} \rightarrow \begin{matrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{matrix}$

Hijo 1: 1010 → P=15, Z=10 ✓

Hijo 2: 1110 → P=21, Z=15 ✗

Probabilidad de primer Padre: 0.70 → Padre 1: 0110

Probabilidad de segundo Padre: 0.35 → Padre 2: 1010

Probabilidad de cruce: 0.26 → $0.26 \leq 0.98$ → Cruzan

Probabilidad de corte: 0.74 → Corta entre el gen 3 y el gen 4

Probabilidades de mutaciones: $\begin{matrix} 0.23 & 0.07 & 0.36 & 0.62 \\ 0.52 & 0.16 & 0.05 & 0.71 \end{matrix}$ → Mutan el gen 2 del padre 1 y el gen 3 del padre 2.

$\begin{matrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{matrix} \rightarrow \begin{matrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{matrix} \rightarrow \begin{matrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{matrix}$

Hijo 3: 0010 → P=8, Z=6 ✓

Hijo 4: 1000 → P=7, Z=4 ✓

Faltaría seguir hasta encontrar un hijo más y armar una nueva población.

Ejercicio 5

Un carguero con capacidad de 300 toneladas puede transportar los siguientes contenedores, maximice el beneficio sin superar la capacidad utilizando Algoritmo Genético. (Aplique el algoritmo hasta obtener una nueva población).

Contenedor	C1	C2	C3	C4
Peso	100	155	50	112
Beneficio	20	35	22	5

$$\text{Max } Z = 20C_1 + 35C_2 + 22C_3 + 5C_4$$

$$100C_1 + 155C_2 + 50C_3 + 112C_4 \leq 300$$

Probabilidad de Cruce: 0.98

Probabilidad de Mutación: 0.1

Población Inicial: 1010 0101 1100 0100

Cromosoma	Z	P	Prob	ProbAc
1010	42	150	0.2441	0.2441
0101	40	268	0.2325	0.4766
1100	55	255	0.3197	0.7963
0100	35	155	0.2034	0.9997
	172			

Probabilidad de primer Padre: 0.43 → Padre 1: 0101

Probabilidad de segundo Padre: 0.69 → Padre 2: 1100

Probabilidad de cruce: 0.84 → $0.84 \leq 0.98$ → Cruzan

Probabilidad de corte: 0.44 → Corta entre el gen 1 y el gen 2

Probabilidades de mutaciones: $\begin{matrix} 0.29 & 0.37 & 0.03 & 0.62 \\ 0.05 & 0.63 & 0.41 & 0.17 \end{matrix}$ → Mutan el gen 3 del padre 1 y el gen 1 del padre 2.

$\begin{matrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{matrix} \rightarrow \begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{matrix}$

Hijo 1: 0110 → P=205, Z=57 ✓

Hijo 2: 0101 → P=268, Z=40 ✓

Probabilidad de primer Padre: 0.70 → Padre 1: 1100

Probabilidad de segundo Padre: 0.35 → Padre 2: 0101

Probabilidad de cruce: 0.26 → $0.26 \leq 0.98$ → Cruzan

Probabilidad de corte: 0.74 → Corta entre el gen 2 y el gen 3

Probabilidades de mutaciones: $\begin{matrix} 0.23 & 0.07 & 0.36 & 0.62 \\ 0.52 & 0.16 & 0.05 & 0.71 \end{matrix}$ → Mutan el gen 2 del padre 1 y el gen 3 del padre 2.

$\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{matrix} \rightarrow \begin{matrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{matrix}$

Hijo 3: 1001 → P=205, Z=25 ✓

Hijo 4: 0110 → P=212, Z=57 ✓

Nueva Población:

Cromosoma	Z	P	Prob	ProbAc
0110	57	212	0.3184	0.3184
0101	40	268	0.2234	0.5418
1001	25	205	0.1396	0.6814
0110	57	212	0.3184	0.9998
	179			

Teorema de Bayes – Regla Bayesiana

Ejercicio 1

Defina: Probabilidad a Priori. Probabilidad Condicional. Expresé el Teorema de Bayes. Defina Red Bayesiana.

Probabilidad a Priori: La probabilidad a priori, también conocida como probabilidad previa o probabilidad prior, se refiere a la probabilidad de que un evento ocurra antes de considerar cualquier evidencia adicional. Es una estimación inicial de la probabilidad de un evento sin tener en cuenta observaciones o datos específicos. Se basa en el conocimiento o la información previa, generalmente subjetiva o teórica.

Probabilidad Condicional: La probabilidad condicional se refiere a la probabilidad de que ocurra un evento dado que otro evento ya ha ocurrido. Se denota como $P(A|B)$, donde "A" es el evento que estamos interesados en y "B" es el evento condicional que ya ha sucedido. La probabilidad condicional tiene en cuenta la información adicional proporcionada por la ocurrencia de "B".

Teorema de Bayes: El Teorema de Bayes es una fórmula fundamental en teoría de probabilidad y estadísticas que permite actualizar las probabilidades a priori en función de nueva evidencia. Se expresa de la siguiente manera:

$$P(b|a) = \frac{P(a|b) * P(b)}{P(a)}$$

$P(A|B)$ es la probabilidad posterior de que ocurra el evento "A" dado que ha ocurrido el evento "B".

$P(B|A)$ es la probabilidad de que ocurra el evento "B" dado que ha ocurrido el evento "A".

$P(A)$ es la probabilidad a priori del evento "A".

$P(B)$ es la probabilidad a priori del evento "B".

Red Bayesiana: Una Red Bayesiana es un modelo gráfico probabilístico que se utiliza para representar y resolver problemas de incertidumbre y toma de decisiones. Consiste en un conjunto de nodos (variables) interconectados que representan eventos o estados, y arcos que representan relaciones de dependencia probabilística entre ellos. Las Redes Bayesianas se basan en la probabilidad condicional y permiten modelar la probabilidad de un evento dado el conocimiento de otros eventos en un sistema. Se utilizan en una amplia gama de aplicaciones, como diagnóstico médico, sistemas de recomendación, planificación y toma de decisiones bajo incertidumbre.

Ejercicio 2

Aplice el Teorema de Bayes para determinar la probabilidad de que un empleado directivo de una empresa de automotores, elegido al azar sea ingeniero. Se cuenta con la siguiente información:

- La empresa tiene empleados que son ingenieros, economistas y empleados de otras carreras.
- El 20% de los empleados de una empresa son ingenieros
- Otro 20% de los empleados son economistas.
- El 75% de los ingenieros ocupan un puesto directivo.
- De los economistas el 50% también ocupa un puesto directivo.
- De los no ingenieros y los no economistas solo el 20 % ocupa puesto directivo.

$$P(A_1) = 0.2 \quad P(A_2) = 0.2 \quad P(A_3) = 0.6$$

$$P(B|A_1) = 0.75 \quad P(B|A_2) = 0.50 \quad P(B|A_3) = 0.2$$

Siendo A_1 = Empleados que son ingenieros, A_2 = Empleados que son economistas, A_3 = empleados que no son ni ingenieros ni economistas y B = Probabilidad que un empleado sea directivo.

$$P(A_1|B) = \frac{P(B|A_1) P(A_1)}{P(B)} = \frac{0.75 \times 0.2}{(0.75 \times 0.2) + (0.5 \times 0.2) + (0.2 \times 0.6)} = 0.4054$$

Ejercicio 3

Dada la siguiente información calcule la probabilidad de que no haya habido ningún incidente, en el supuesto de que haya funcionado la alarma:

- La probabilidad que haya un accidente en una fábrica que dispare la alarma es 0.1
- La probabilidad de que suene ésta si se ha producido algún incidente es de 0.97
- La probabilidad de que suene si no ha sucedido ningún incidente es 0.02

Prob de accidente que dispare la alarma 0.1

Prob de no accidente que dispare la alarma 0.9

Prob de que suene la alarma con algún accidente 0.97

$P(a|b) = 0.02$ = Prob de que suena la alarma sin accidente

$P(b|a)$ = Prob que no haya accidente cuando se dispara la alarma

$$P(b|a) = \frac{P(a|b) P(b)}{P(a)} = \frac{0.02 \times 0.9}{(0.97 * 0.1) + (0.02 * 0.9)} = \frac{0.018}{0.115} = 0.1565$$

Ejercicio 4

Ciertos medicamentos y traumas pueden causar coágulos de sangre. Un coágulo de sangre puede provocar un ACV (Accidente Cerebro Vascular), un ataque cardíaco o simplemente podría disolverse por sí solo y no tener implicaciones para la salud.

- Crear un DAG (Gráfica Acíclica Dirigida).
- Dada la siguiente información, ¿cuál es la probabilidad de que una persona desarrolle un coágulo de sangre como resultado tanto de la medicación, como del trauma y entonces no tenga implicaciones médicas?

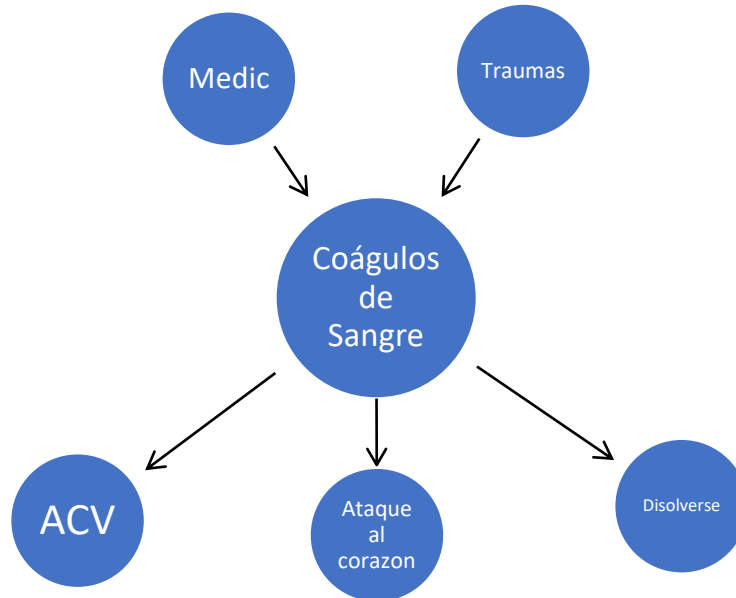
Donde M = medicación, T = trauma, BC = coágulo de sangre, HA = ataque cardíaco, N = nada y S = ACV (accidente cerebrovascular). T significa verdadero, o este evento sí ocurrió. F significa falso, o este evento no ocurrió.

$P(M=T)$	0.2
$P(M=F)$	0.8
$P(T=T)$	0.05
$P(T=F)$	0.95

M	T	$P(BC=T)$	$P(BC=F)$
T	T	0.95	0.05
T	F	0.3	0.7
F	T	0.6	0.4
F	F	0.9	0.1

BC	$P(HA=T)$	$P(HA=F)$	$P(S=T)$	$P(S=F)$	$P(N=T)$	$P(N=F)$
T	0.4	0.6	0.35	0.65	0.25	0.75
F	0.15	0.85	0.1	0.9	0.75	0.25

Ítem a)



Ítem b)

$$P(N \wedge BC \wedge T \wedge M) = P(N|BC) \cdot P(BC|T \wedge M) \cdot P(T) \cdot P(M) = 0.25 \cdot 0.95 \cdot 0.2 \cdot 0.05 = 0.002375$$

Ejercicio 5

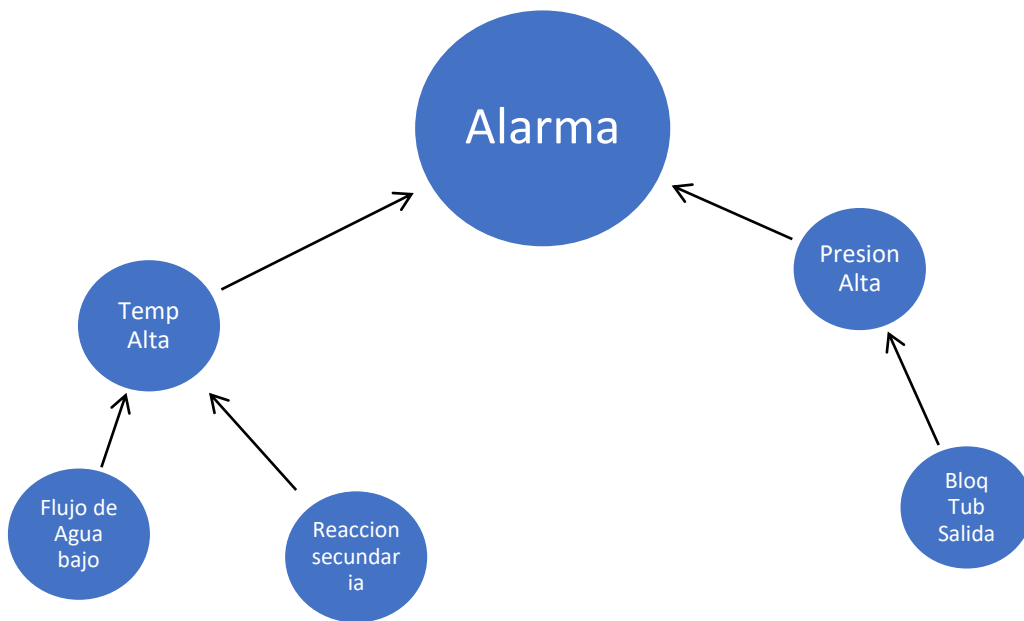
Una alarma (A) multipropósito en una planta se puede activar de dos maneras. La alarma (A) se activa si la temperatura del reactor es demasiado alta o la presión en un tanque de almacenamiento es demasiado alta.

- La temperatura del reactor puede ser demasiado alta debido a un flujo de agua de enfriamiento (CFL) bajo (1% de probabilidad) o por una reacción secundaria (SR) desconocida (5% de probabilidad).
- La presión del tanque de almacenamiento podría ser demasiado alta debido a un bloqueo en la tubería (PB) de salida (2% de probabilidad).
- Si el flujo de agua de enfriamiento (CFL) es bajo y hay una reacción secundaria (SR) entonces hay una probabilidad del 99 % de que ocurra una temperatura alta (HT).
- Si el flujo de agua de enfriamiento (CFL) es normal y no hay reacción secundaria (SR), solo hay un 3% de probabilidad de que ocurra una temperatura alta (HT).
- Si hay un bloqueo de la tubería, siempre se producirá alta presión (HP).
- Si no hay obstrucción de tuberías (PB), se producirá una alta presión (HP) solo el 2% del tiempo.

- Crear una DAG para la situación anterior.
- Configurar las tablas de probabilidad necesarias para modelar este sistema. No se dan todos los valores requeridos para llenar estas tablas, por lo tanto rellene lo que sea posible.

CFL = El flujo de agua fría es bajo, SR = Reacción Secundaria, PB = Tubería bloqueada, HT = Alta temperatura, HP = Alta presión, A = Alarma. T significa verdadero, o el evento sí ocurrió. F significa falso, o el evento no ocurrió.

Ítem a)



Ítem b)

$HT CFL$	0.01
$HT \neg CFL$	0.99
$HT SR$	0.05
$HT \neg SR$	0.95

$HP PB$	0.02
$HP PB$	0.98

CFL	SR	HT=T
T	T	0.99
F	F	0.03

PB	HP
T	1.00
F	0.02