

Ejemplo de preguntas de examen TIA

- 1) ¿Qué ventajas tiene un razonamiento hacia atrás (BC) frente a un razonamiento hacia adelante (FC) en un SBC desde el punto de vista computacional? ¿Qué requisitos concretos tiene?

Ventajas BC frente FC: se centra en la información más relevante para el objetivo, permite preguntar al usuario y, en general, suele ser más eficiente. Además, FC genera más información.

Características razonamiento hacia atrás:

- Guiado por los objetivos. Más parecido a la conducta humana (eficacia).
- Necesita objetivos explícitos, que intenta cumplir.
- Obtiene respuestas a preguntas.
- Permiten preguntar al usuario.
- Control importante: Orden preguntas, satisfacción subobjetivos (eficacia).

Características razonamiento hacia adelante:

- Guiado por los Datos.
- Sistemas generativos: Obtiene todos los hechos deducibles.
- Control no importante si: Monótono (no retracta) y todas las acciones son relevantes.

- 2) Explica breve, pero razonadamente, cuáles son los elementos principales de un Sistema Basado en el Conocimiento (SBC).

Los elementos principales serían:

- Conocimiento del Dominio del Problema (Base de Conocimiento).
- Datos del problema (Base de Hechos).
- Resolución de problemas (Motor de inferencia).
- Interfaz de desarrollo y usuario.

Alternativamente:

- Base de conocimiento.
- Mecanismos de razonamiento.
- Memoria de trabajo.

- 3) Define los conjuntos y reglas difusas necesarias para expresar la siguiente información:

"Las personas que ganan alrededor de 30.000 euros/año tienen un coche pequeño"

"Las personas que ganan mucho dinero tienen un coche grande"

En base a la anterior representación, indica cómo podría obtenerse el tamaño del coche de Juan, que gana 35.000 euros/año. Esquematiza el proceso a seguir, no hace falta que se realicen los cálculos con detalle.

~ Conjuntos ~

dinero: {poco, bastante, mucho}

coche: {pequeño, mediano, grande}

dinero mucho: 30.000 35.000 / -

dinero bastante: 20.000 25.000 30.000 35.000 / - \

dinero poco: 20.000 25.000 - - \

~ Reglas ~

dinero: bastante -> coche: pequeño

dinero: mucho -> coche: grande

35.000€/año -> fusificamos -> dinero: mucho -> por tanto -> coche: grande

supongo xxxxxxxx A

- 4) Un sistema basado en reglas con un mecanismo hacia delante genera la misma información siguiendo una estrategia en amplitud/anchura que en profundidad? Explícalo razonadamente y pon un ejemplo ilustrativo de ello.

No generaría la misma información, porque si lo viésemos cómo un árbol, en el caso de amplitud generaría todos los nodos de cada nivel del árbol, y en el caso de profundidad generaría nodos hijo hasta llegar a las hojas o encontrar la solución, en el caso de no encontrarla, habría que hacer backtracking y recorrer otro camino.

- 5) Explica razonada y detalladamente para qué es útil la toma de decisiones con incertidumbre. Pon un ejemplo de ello e indica brevemente los pasos que realizarías para resolver dicho ejemplo.

Es útil debido a que en el mundo real los eventos no tiene probabilidades de 0 o 1 hasta que son evidenciadas, sino que poseen una cierta incertidumbre, alejándose de estos valores, y en general suelen estar condicionadas por distintos factores.

Porque se desconocen parte de los datos, son cualitativos y/o simbólicos, no muy precisos, etc. Por tanto, se deben de tomar decisiones en base a información dinámica, incierta, incompleta, imprecisa y contradictoria, por lo que se necesita ampliar la lógica (modelo con incertidumbre) para tratar y representar dicha información.

- 6) Si tuvieras que tomar una decisión ante la situación “subo la apuesta o me retiro”, explica razonadamente qué criterios seguirías y en qué te basarías para decidirte. Indica si es una decisión racional y por qué.

La decisión se tomaría en cuenta según la probabilidad de ganar y la cantidad a qué subir la apuesta. Habría que valorar primero qué probabilidad hay de ganar y qué de perder, para valorar tres casos: subir la apuesta y ganar, subir la apuesta y perder, y retirarse, suponiendo que ésta última suponga una pérdida de lo apostado actualmente. Esto por tanto es una decisión racional ya que comporta asociar una función de utilidad al dinero que se va a poder ganar (o perder).

Si decides no arriesgarte pese a que la opción de apostar maximiza la función de utilidad es por la aversión al riesgo, algo inherente en las acciones humanas, por tanto no sería una decisión racional.

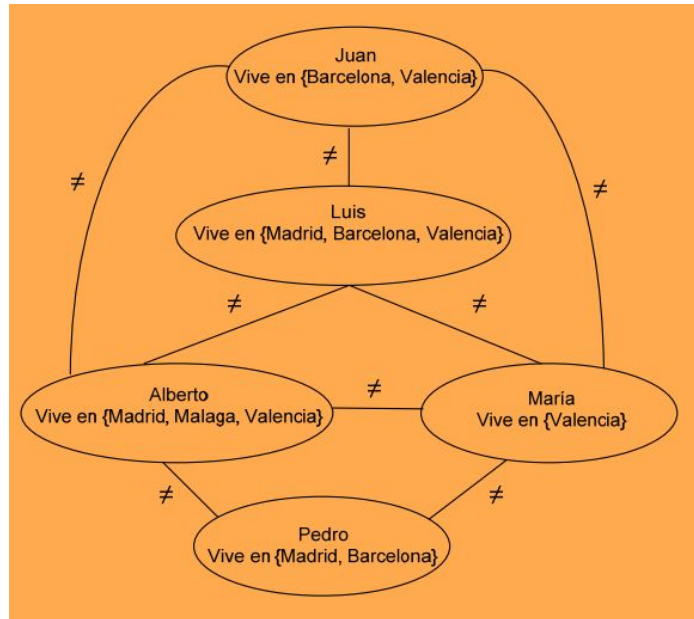
- 7) En teoría de juegos, y considerando un juego de dos jugadores ¿Qué es un punto de silla? Supóngase la siguiente matriz de pagos, con los jugadores A y B, que pueden elegir las alternativas 1,2 o 3, ¿Existe un punto de silla?

	Jugador-A: 1	Jugador-A: 2	Jugador-A: 3
Jugador-B: 1	Jug-B: 2, Jug-A:8	Jug-B: 1, Jug-A:9	Jug-B: 9, Jug-A:1
Jugador-B: 2	Jug-B: 4, Jug-A:6	Jug-B: 5, Jug-A:5	Jug-B: 6, Jug-A:4
Jugador-B: 3	Jug-B: 3, Jug-A:7	Jug-B: 8, Jug-A:2	Jug-B: 7, Jug-A:3

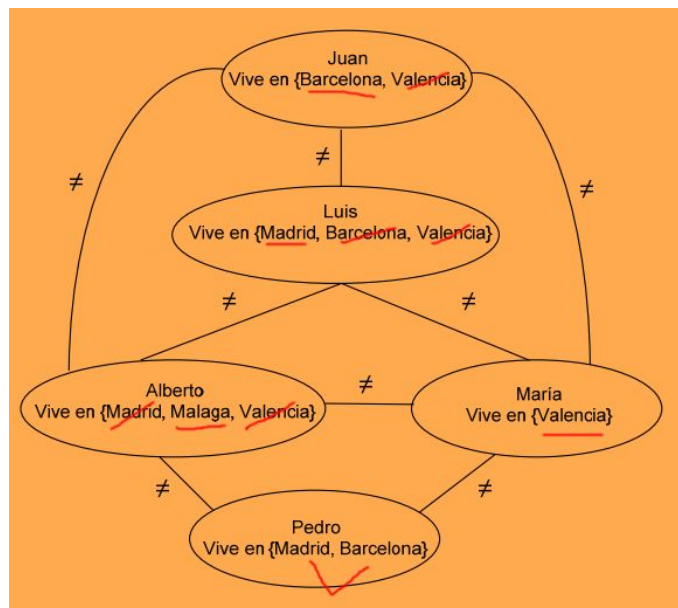
Un punto de silla es un par de estrategias donde ningún jugador puede mejorar cambiando su estrategia.

Existe el punto de silla (A: 1, B: 2)

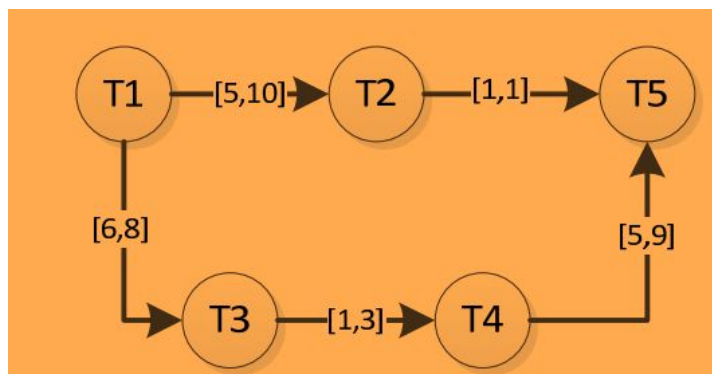
- 8) Qué utilidad tiene un proceso inferencias en un CSP? (contestar concreta y brevemente)
Aplicar un proceso inferencial 'arco-consistencia' al siguiente CSP y obtener el CSP resultante:



Un proceso de inferencias limita los dominios de las variables y las restricciones entre ellas



- 9) Aplica un proceso inferencial de clausura transitiva (senda consistencia o path-consistency) a la siguiente red temporal que representa un CSP donde las variables son instantes de tiempo T1..T5. Razona si la red clausurada es consistente.



$T1 \rightarrow T2 \rightarrow T5: [5, 10] \times [1, 1] = [6, 11]$

$T1 \rightarrow T3 \rightarrow T4: [6, 8] \times [1, 3] \times [5, 9] = [12, 20]$

La intersección de estos dos resultados (intervalos) resulta nula, por lo que podemos afirmar que la red clausurada NO es consistente.

- 10) ¿Qué es un CSP flexible? Pon un ejemplo de un CSP flexible indicando el tipo de CSP flexible elegido.

Un CSP flexible es aquel que utiliza restricciones blandas (no imperativas y flexibles) para poder resolver problemas de optimización. Además, se incluyen funciones de coste que expresan el grado de satisfacción de cada tupla. Un ejemplo de un CSP flexible posibilista (algunas restricciones no se cumplen, su objetivo es minimizar el coste de las restricciones no satisfechas) sería aplicar costes de insatisfacibilidad en el problema de las n-reinas.

- 11) Suponiendo que queremos resolver el juego del Sudoku, de 9x9, mediante un AG, diseña dos posibles individuos de la población y aplicarles un proceso de cruce y mutación para obtener dos nuevos individuos. Define una posible función fitness (no hace falta especificar matemáticamente)

[..., 7, 3, 6, 2] & [..., 5, 8, 1, 3]

Cruce:

[..., 7, 3, | 6, 2] & [..., 5, 8, | 1, 3] => [..., 5, 8, 6, 2] & [..., 7, 3, 1, 3] (no válido: no puede repetirse el mismo número (en este caso el 3) en una fila -> fitness -infinito)

)

Mutación (por inserción):

[..., 5, 8, 6, 2] => [..., 5, 2, 8, 6]

- 12) Desde el punto de vista de representación, ¿en qué se parece/diferencia una acción PDDL a una regla de un SBC? Razona la respuesta.

La estructura es similar, pero se diferencian en que en PDDL las acciones pueden requerir la asignación de recursos (renovables o no), con información numérica que modelan capacidades, costes.

- 13) ¿Qué objetivo tiene un proceso inferencial en un CSP? ¿Cuál es su utilidad y para qué sirven? ¿Son útiles para la búsqueda de una solución? Razona la respuesta.

- Encontrar una solución consistente (que satisfaga todas las restricciones).
- Los csp se utilizan para resolver problemas con restricciones.
- Sirven para problemas en los que se busca una o más soluciones satisfaciendo todas las restricciones.

- 14) En la resolución de un Problema de Satisfacción de Restricciones, ¿se aplica, o se puede aplicar conocimiento heurístico? ¿Dónde?

Se pueden aplicar heurísticas en las técnicas de búsqueda para ordenar los valores o variables.

- 15) El problema del cartero chino consiste en encontrar el camino más corto en un grafo que pase al menos una vez por cada arista del mismo, volviendo a la posición (nodo) de partida. Es decir, se trata de que el cartero visite todas las calles para poder realizar el reparto de correo, siendo la oficina de correos su punto de partida y llegada. Diseña el problema mediante un algoritmo genético. Para ello hay que: 1) proponer una codificación de los individuos, 2) definir la función fitness de evaluación y 3) indicar dos individuos de la población inicial. Detalla los procesos de codificación y decodificación de los individuos para aplicar la función fitness definida.

Codificación y dos individuos de la población inicial:

[1, 2, 4, ..., 7, 5, 1] & [1, 5, 6, ..., 2, 3, 1] -> Suponiendo que la oficina de correos es el 1 y hay x calles

Evaluación:

Si no cumplen la restricción de que se han recorrido todas las aristas, y que el último y primer nodo sean el 1, se les proporciona un valor de -Infinito, en el caso contrario, el valor del fitness es la suma de los pesos de cada arista que hay entre cada par de números. Para la decodificación, se podría entender que tendríamos una matriz en la que las filas y columnas representan los nodos, y el valor es el coste de la arista que une esos dos nodos, si es -1 significa que no existe la arista.

- 16) ¿Explica razonadamente qué ocurriría en un algoritmo genético que no implementara el operador de cruce de individuos? ¿Qué ocurriría si no se implementa el operador de mutación? ¿Puedes indicar cuál de las dos opciones encontraría antes la solución óptima y si hay garantía de encontrarla?

Si no implementase el operador cruce no se obtendrían individuos con características de dos progenitores, por lo que la diversidad genética de la población se vería drásticamente reducida. Ocurriría algo similar con las mutaciones pero en menor grado. En este último caso a corto plazo no sería muy negativo, pero a largo plazo daría lugar a una convergencia, pues llegaría un momento en el que todos los posibles cruces ya se habrían dado y por tanto no habría evolución ni diversidad genética.

La solución óptima se encontraría antes con solo la operación de cruce, ya que es primordial ser capaz de generar hijos. No existe garantía de encontrarla pero en ningún caso ya que en general los algoritmos genéticos están acotados por el número de soluciones, cómputo, etc. no ofreciendo siempre la solución óptima, porque puede no haberse hallado.

En resumen, un algoritmo sin cruce y con mutación sería demasiado explorado pero poco explotado, y un algoritmo con cruce y sin mutación sería demasiado explotado y poco explorado.

- 17) Deseamos modelar un escenario de planificación en PDDL para gestionar dos grúas móviles que se dedican a la carga y descarga de palets de camiones. Cada grúa puede realizar las siguientes acciones:

- Cargar un palet al camión y descargar un palet del camión. Las acciones de carga/descarga tienen un coste de 5 unidades cada una y una duración de 10 unidades.
- Moverse de una posición a otra. Tanto el coste como la duración dependen de la distancia entre las dos posiciones. Los valores se dejan a libertad del alumno.

Definir el modelo PDDL durativo de planificación, tanto el dominio como un breve problema de ejemplo (no hace falta definir el objetivo/goal del problema, basta con definir una información inicial de partida).

(:durative-action load-truck ;Load 'cargo' in 'truck' with 'crane', (grúa puede simultanear cargas)

```
:parameters      (?t - truck)
                  (?l - location)
                  (?o - cargo)
                  (?c - crane)

:duration          (= ?duration 5)

:condition (and
  (at start (at ?t ?l))      ; truck en ?l al inicio
  (over all (at ?t ?l))      ; truck en ?l overall
  (at start (at ?o ?l))      ; cargo en ?l
  (at start (empty ?c)))     ; grúa disponible (y compatible)

:effect (and
  (at start (not (at ?o ?l))) ; cargo no en ?l
  (at end (in ?o ?t)))        ; cargo en truck
```



Si se quiere especificar que una condición <p> se mantiene en el intervalo cerrado durante la duración de una acción durativa, entonces se requieren tres condiciones:

(at start <p>), (over all <p>) y (at end <p>)



- 18) Estamos ante un problema de planificación que dispone de 4 acciones: ponerse calcetín derecho, ponerse calcetín izquierdo, ponerse zapato derecho y ponerse zapato izquierdo. Obviamente, antes de ponerse el zapato de un pie es necesario ponerse el calcetín de ese mismo pie. Suponiendo que la duración de estas 4 acciones es unitaria, dibuja el grafo de planificación asociado y contesta a las siguientes preguntas sobre dicho grafo:

- Suponiendo que podemos ponernos los dos zapatos a la vez, ¿cuántos niveles de acción son necesarios para conseguir el objetivo de tener puestos los dos zapatos? Indica en este caso qué relaciones de exclusión mutua (mutex) aparecen.
- Suponiendo que disponemos de un único calzador (que impide ponerse los dos zapatos a la vez), ¿cuántos

niveles de acción son necesarios para conseguir el objetivo de tener puestos los dos zapatos? Indica en este caso qué relaciones de exclusión mutua (*mutex*) aparecen.

NOTA: en todos los casos podemos suponer que somos capaces de ponernos los dos calcetines al mismo tiempo.

a) Habrían 2 niveles de acción: ponerse los dos calcetines al mismo tiempo, ponerse los dos zapatos al mismo tiempo. En este caso no habrían acciones mutex, porque en ningún nivel de acción hay dos acciones que no se puedan ejecutar al mismo tiempo.

b) Habrían 3 niveles de acción: ponerse los dos calcetines al mismo tiempo, ponerse un zapato derecho o izquierdo, ponerse el zapato contrario. Son acciones mutex ponerse el zapato derecho y ponerse el zapato izquierdo ya que no se pueden realizar a la vez.

19) Diseña un modelo basado en restricciones (CSP) para resolver el juego del Sudoku. Hay que modelar las variables, sus dominios y las restricciones involucradas.

Esquema:

Sudoku:

rellenar las celdas con números del 1..9,
tal que no se repitan en columnas, filas o submatrices.

	2	9						
				3		1		
8			4					
9			8	2				5
	5				6	9		7
					5			
		4			1			6
		8	9					2
2	1					5		

Formulación:

- **variables:** 9x9 celdas
- **dominios:** {1..9}
- **restricciones:** propiedades que se deben satisfacer:
 - Todas las variables de una submatriz: distintas
 - Todas las variables de una fila: distintas
 - Todas las variables de una columna: distintas

CSP no binario, con restricciones disyuntivas