1. Explique Sistema Experto y su estructura
2. Explique Unificación y Correspondencia (ligadura) -🡪 no se de donde copiaron esta pero ligadura va relacionado con Unificación y no con Correspondencia
3. Definición de Clausula de Horn y Tipos (Por tipos interpreto que pregunta los tipos de Tipos de formas clausales)
4. Explique el funcionamiento de la pila de objetivos.
5. Explique las etapas de Lógica Difusa.
6. ventajas y desventajas en métodos de búsqueda no informados (quizás podríamos poner mas info).
7. Heurística
8. Procedimiento mediante pila de objetivos (Idem a la 4 creo)
9. Lógica de Predicados. Lógica Proposicional. Formas Clausulas de HORN, Importancia de Cláusulas de Horn.
10. Explique el concepto de función de membrecía en l.....Enumere y explique al menos dos operadores Difusos

De otra recopilación:

1. Característica de LISP.
2. Tipos de Planificación (Algo mas para poner?)

1)

Es un programa destinado a generar inferencias en un área específica del conocimiento en una forma similar a la que se espera de un experto humano.

Estos sistemas resuelven problemas por aplicación de conocimientos en un dominio específico. Este conocimiento es adquirido a través de la intervención de expertos humanos y almacenados en lo que se denomina BD de conocimientos (KDB).

Se espera que un sistema experto sea capaz de explicar la solución hallada de forma comprensible para las personas. Además no pueden operar en situaciones de sentido común por ser muy extenso el dominio de conocimiento que debe tener el sistema.

**Características del tipo de problema resuelto por un SE:**   
- Inexistencia de algoritmo para hallar solución.  
- Problemas resueltos por expertos satisfactoriamente.  
- Contar con un experto humano para su desarrollo.  
- El conocimiento experto del dominio debe ser relativamente estático.

**Aplicaciones:** Clasificación, interpretación, predicción, diagnostico, monitoreo, diseño, etc. Pueden ser aplicados a mundos de muy baja complejidad.

**Arquitectura (Intrepreto que es lo mismo que estructura):**   
1- El núcleo de un SE es el motor de inferencia  
2- Reglas a aplicar sobre los símbolos definidos.   
 - Se obtienen del conocimiento del experto humano.  
 - Se combinan con un sistema de inferencias.  
 - Se consideran heurísticas, por no poder demostrar su validez general.  
3- Hechos. Predicados que se suponen verdaderos. Constituyen el KDB.  
La función del ingeniero del conocimiento proporcionar el conocimiento para poder construir la base de conocimiento y el motor de inferencias

EXPLICACIONES

MOTOR DE INFERENCIAS

KDB

EDITOR

INGENIERO

USUARIO

INTERFACE

2)

UNIFICACIÓN Y LIGADURA

Permite razonar dentro de un esquema formal sin relación con la codificación de los programas. En el caso de un código, este se encarga de reducir las formas clausales a listas asociadas más simples de manipular denominadas ligaduras, esta fluye a través de los filtros del programa y al final del proceso pueden ser analizadas para extraer nuevas afirmaciones o demostrar hipótesis.

CORRESPONDENCIA DE PATRONES

Relacionado con la codificación de SE es la correspondencia de patrones simbólicos. La correspondencia con patrones se basa en identificar si dos vectores son similares es decir se corresponden. La forma simple de entenderlo es calcular la distancia entre los mismos y si esta es lo suficientemente pequeña de acuerdo a valores preestablecidos para un problema dado, entonces puede decirse que ambos vectores se corresponden.

Correspondencia Simbólica

Busca similitudes entre un conjunto de nombres de variables y no entre los valores numéricos de estas. Se consideran dos procedimientos:

**Correspondencia:** encadenamiento hacia adelante, comparando una expresión común (dato) con un patrón. A diferencia de los datos los patrones pueden contener variables patrón, para facilitar el reconocimiento de estas variables se asocian a un símbolo.

**Unificación:** encadenamiento hacia atrás. Hace corresponder dos patrones en lugar de un patrón y un dato.

Correspondencia Sub simbolica

Correspondencia numérica.

3)

Una clausula de Horn es una conjunción de disyunciones como no más de un literal positivo. Entendiendo como un literal positivo a uno que no esté negado. Por ejemplo: (-A1 v A2) ^ (–B1 v –B2 v –B3).

Estas cláusulas de Horn representan la forma implicativa escrita de otro modo, es decir, la implicación B 🡪 A equivale a –B v A y la implicación B1 ^ B2 🡪 A equivale a –B1 v –B2 v A.

Para aplicar la lógica en programas de computadora ha sido necesario encontrar algoritmos que determinen si una expresión lógica válida se hace verdadera con alguna combinación de valores verdaderos o falsos de sus átomos. Un algoritmo conocido pero de altísimo costo es construir la tabla de verdad de la expresión y así determinar si alguna combinación hace verdadera la expresión. Un algoritmo más rápido puede aplicarse si la expresión está en la forma de cláusula de Horn, y aquí radica su importancia.

Tipos de formas clausales

- Forma Normalizada Conjuntiva: (A v –B) ^ (B v C) ^ A. Conjunción de disyunciones de literales.

- Forma Normalizada Disyuntiva: (–P ^ – Q) v (–Q ^ R) v P. Disyunción de conjunciones de literales.

- Cláusula de Horn: La cláusula contiene como máximo un literal positivo.

4)

(Primero doy una introducción pero no es necesaria)

La planificación se refiere al proceso de computar varios pasos de un procedimiento de resolución de un problema antes de ejecutar alguno de ellos.

Cuando nos enfrentamos con el problema de planificar y ejecutar una secuencia de pasos en un mundo que no es completamente predecible nos enfrentamos al problema de planificación en IA.

Clasificación:

Planificación en IA:

* Numérica
* Lógica: Por pila de objetivos, No lineal, Jerárquica
* Reactiva

Planificación en robótica

* Reactiva
* De trayectoria de manipuladores

Planificación industrial: métodos basados en simulación discreta

Funcionamiento de la Pila de Objetivos

**Elementos:** Bloques, superficie y brazo operador.

**Estados de los elementos:** estado del bloque y del brazo.

**Operadores:** acciones a realizar. Se componen de tres listas: precondición (lo que se debe cumplir para aplicar el operador), Borrado (lo que se debe eliminar de la pila de objetivos), Adición (lo que se agrega al plan de tareas)

**Pilas:** en todo momento se cuanta con una pila de objetivos y operadores, una de estado actual y la del plan a seguir.

**Procedimiento:** se tratan los objetivos como sub problemas a resolver. Se pregunta si el primer objetivo se cumple. Si no se cumple se debe aplicar el operador que lo transforme, por lo que se debe reemplazar al predicado objetivo por el operador y luego agregar a la pila las precondiciones que deben cumplirse para ese operador. Para resolver el objetivo que ha quedado arriba de la pila empleamos el operador reemplazando y agregando las precondiciones. Al ir extrayendo los operadores de esa pila, se incorporan a la lista plan y se actualiza el estado actual.

5)

1. Difusión de las entradas - Fuzzificacion

Se deben resolver todas las sentencias en el antecedente en función de su grado de membresía entre 0 y 1. Si el antecedente tiene solo un componente este es el grado de soporte del consecuente.

Tomar las entradas y determinar el grado en el cual ellos pertenecen a cada uno de los conjuntos difusos (adjetivos) a través de funciones de membresía o pertenencia –evaluación de una función-.

La entrada es siempre un valor numérico limitado al universo de discurso de la variable de entrada y la salida es un grado difuso de pertenencia.

2. Aplicación de operadores difusos

Si existen múltiples partes en el antecedente se aplican los operadores fuzzy y se resuelve el antecedente como un número entre 0 y 1. Esto es el grado de soporte de la regla.

En este paso ya se conoce el grado en el cual cada parte del antecedente ha sido satisfecho para cada regla. Pero, como se menciono anteriormente, si el antecedente de una regla ha sido dividido en más de una parte, se debe aplicar un operador difuso para obtener un número que representa el resultado del antecedente para esa regla.

Hay distintos métodos u operadores fuzzy: mínimo, producto, truncamiento; máximo, amplificación y adición.

3. Aplicación de la implicación

Se emplea el grado de soporte de la regla para conformar el conjunto fuzzy de salida. El consecuente de una regla fuzzy asigna un conjunto fuzzy completo a la salida. Si el antecedente es solo parcialmente cierto, luego el conjunto fuzzy de salida es truncado según el método de implicación. Es decir, es la conformación del consecuente (conjunto fuzzy) basado en el antecedente (un número). Se obtiene una salida para cada regla por separado.

4. Agregación

En general se requieren dos o más reglas que interactúen entre ellas. La salida de cada regla es un conjunto fuzzy, pero en general, se necesita que la salida sea en definitiva un número. Para obtener este número, primero los conjuntos fuzzy se salida de cada regla se amalgaman en un único conjunto. Luego este conjunto es defuzzificado, o resuelto de modo de obtener un número como valor de salida.

Este paso entonces, es el proceso de unificar las salidas para cada regla uniendo los procesos paralelos. Se crean un único conjunto difuso a partir de las salidas truncadas o modificadas retornadas por el proceso de implicación para cada regla.

5. Defuzzificacion o desdifusión

La entrada para el esta etapa es el conjunto difuso agregado (que engloba el rango de valores de salida) y la salida es un número. Un popular método de defuzzificacion es el cálculo del centroide, el cual retorna el centro de un área bajo una curva.

La defuzzificacion tiene lugar en dos pasos:

1. Las funciones de pertenencia son escaladas de acuerdo a sus posibles valores.

2. Estas son usadas para calcular el centroide de los conjuntos difusos asociados.

6)

Primero en anchura BPA

**Ventajas:** No entra en callejones sin salida, si existe una solución garantiza alcanzarla, si existen varias soluciones garantiza alcanzar la optima.

**Desventajas:** utiliza mucha memoria, es lento, mala performance.

Primero en profundidad BPP

**Ventajas: S**i existe una solución garantiza alcanzarla, no recorre todo el árbol para alcanzar la solución, pero es por azar.

**Desventajas:** La segunda ventaja es por azar, no es una característica confiable, pudiendo suceder lo contrario.

Ramificación y Acotación (Es el bpp acotado creo, cuya caracteristica es la siguiente, creo)

La diferencia entre la búsqueda primero en anchura (BPA) y primero en profundidad (BPP) es que la BPP hace un descenso por la primera rama del árbol y la BPA recorre el árbol en forma horizontal

7)

Es una técnica que evita la explosión combinatoria en un proceso de búsqueda, posiblemente sacrificando demandas de completitud. Es una buena solución, pero no siempre es la óptima. La heurísticas son buenas aproximaciones que están ligadas a la comprensión del dominio del problema.  
Una función heurística asigna valores de bondad a cada nodo del árbol de búsqueda, para incorporar el conocimiento heurístico.

8) Repetida, igual a la 4

9) Lógica de Predicados y Lógica Proposicional: El instrumento fundamental de la comunicación humana es el lenguaje, formado por frases del tipo: \*Interrogativo, \*Imperativo, \*Declarativo. Las frases declarativas constituyen el elemento básico de descripción del conocimiento. La lógica es la disciplina que estudia los métodos de formalización del conocimiento humano. Por lo tanto la lógica estudia métodos de formalización de frases declarativas. La Lógica de Predicados estudia las frases declarativas con mayor grado de detalle, considerando la estructura interna de las proposiciones. Toma como elementos básicos los objetos y sus relaciones. Es decir: \*Que se afirma (predicado o relación) \*De quien se afirma (objeto) Sócrates es hombre y todos los hombres son mortales, entonces Sócrates es mortal. La Lógica Proposicional se caracteriza en relación a su composición por emplear muy pocos símbolos que pueden ser evaluados solamente como falsos o verdaderos (mediante clausulas y/o elementos atómicos), en contraposición al alfabeto lógico usado por la lógica de predicados para expresar sus formulaciones. \*La Lógica Proposicional está basada en constantes: \*La Lógica de Predicados está basada en variables: .

10) Función de Membresía o Pertenencia: es una función que representa en qué nivel o medida una variable de entrada pertenece al conjunto difuso (siempre entre 0 y 1). Esta es la que le dará forma a los conjuntos difusos. Por ejemplo, si consideramos el conjunto de personas jóvenes y considerando que pertenecen a este grupo los menores de 20 años. Según la lógica tradicional el conjunto se define de la siguiente manera: Joven(x) = 1 si edad (x) <= 20

Joven(x) = 0 si edad (x) > 20 Para la lógica difusa, podemos decir que una persona de 21 años pertenece al conjunto de jóvenes pero con un grado de 0,9. Por lo tanto la función de membresía podría ser: μ joven(x) = 1 si edad(x) <= 20

1 – ((edad(x) -20)/10)

Ahora el conjunto joven contiene a personas entre 20 y 30 años con un grado de membresía que decrece linealmente (DIBUJO). Las funciones de membresía están usualmente restringidas a ciertas clases de funciones. Las funciones pueden ser: (4 DIBUJOS).

Operadores Difusos: Si el antecedente de una regla ha sido dividido en más de una parte se debe aplicar un operador difuso para obtener un número que representa el resultado del antecedente.

Las entradas a un operador difuso son dos o más valores de pertenencia desde las variables de entrada fuzificadas. La salida es un único valor de verdad. Existen diferentes métodos que en general son:

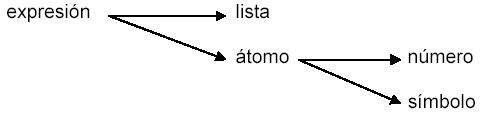
􀁸 Conjunciones: mínimo, producto y truncamiento

􀁸 Disyunciones: máximo, amplificación y adición

􀁸 Negación.

11) LISP

Programa interprete basado en el lenguaje funcional, no genera daños colaterales, es un hibrido ya que me permite realizar bucles y asignar variables. El nombre de LISP surge de abreviar “List Procesing” o Procesamiento de Listas. Fue propuesto por John Mc Carthy en 1959 en el MIT. En LISP toda la información (inclusive las sentencias) se expresan como listas o como los componentes de las mismas.

Un código de LISP está compuesto de formas o expresiones; el intérprete LISP lee una expresión, la evalúa e imprime el resultado. Este procedimiento se denomina ciclo de lectura, evaluación, impresión (read – eval – print loop).

Una **expresión o forma** es una lista o un átomo (símbolo, entero o cadena).

Un átomo es un símbolo o un número. Si la forma es una lista LISP trata el primer elemento como el nombre de una función, y evalúa los restantes elementos recursivamente, y luego llama a la función con los valores de los elementos restantes.

Un **número** se puede escribir en cualquiera de las notaciones habituales.

Un **símbolo** se representa por un nombre que está formado por caracteres alfanuméricos que no puedan interpretarse como números.

Una **lista** es una sucesión de cero o más expresiones entre paréntesis. El blanco (no la coma) es el separador de expresiones dentro de una lista. Cada una de estas expresiones se denomina elemento de la lista.

Cuando LISP evalúa una expresión, devuelve un valor (que será otra expresión) o bien señala un error.

- Un número se evalúa a sí mismo.

- Un símbolo se evalúa al valor que tiene asignado.

- Una lista se evalúa independientemente una por una del siguiente modo, siempre en función del primer elemento y en función de este considera como emplear los elementos siguientes:

1) **El primer elemento** de la lista debe ser un símbolo “s” cuyo significado funcional “F” esté definido.

2) Se evalúan los restantes elementos de la lista, obteniendo los valores v1,…,vn. Si el número o tipo de los valores obtenidos no es coherente con los argumentos requeridos por F, se produce un error.

3) La lista se evalúa a F(v1,…,vn).

La forma en la que se evalúan las listas corresponde a una notación pre fija, donde primero se escribe el operador y luego los parámetros.

FUNCIONES: Existen dos símbolos (T y NIL) que se evalúan a sí mismos. Estos se utilizan como TRUE y FALSE o como TRUE y NULL en otros lenguajes. Un símbolo “s” puede estar ligado, es decir, contener una referencia o indicación a otra expresión “V”. Llamamos ligadura al par (s,V). Un símbolo “s” (salvo T o NIL) se evalúa al valor “V” al que está ligado. Recordar que forma es una expresión que se puede evaluar.

**QUOTE:** Es una forma que tiene un solo argumento. (QUOTE expresión) El valor de (QUOTE expresión) es precisamente “expresión”, sin evaluar. La abreviatura de QUOTE es “ ’ ”. (QUOTE expresión) = ‘expresión Ejemplo: (+ 3 4) 7, ‘(+ 3 4) (+ 3 4), (1 2 3) error, ‘(1 2 3) (1 2 3)

**PREDICADOS**

Son las funciones que devuelven T o NIL.

(EQ e1 e2) es T cuando e1 y e2 son el mismo símbolo

(EQL e1 e2) es T cuando e1 y e2 son EQ o números iguales de igual tipo

(EQUAL e1 e2) es T cuando e1 y e2 son EQL o listas iguales

(= e1 e2 … en) es T cuando e1, e2, …, en son números todos iguales de cualquier tipo

(NULL e) es T cuando e es NIL

(ATOM e) es T cuando e es un átomo

(SYMBOLP e) es T cuando e es un símbolo

(NUMBERP e) es T cuando e es un número

(LISTP e) es T cuando e es una lista

(ENDP e) es T cuando e es NIL (lista vacía). Si e no es una lista da error

(ZEROP e) es T cuando e es cero

(PLUSP e) es T cuando e > 0

(MINUSP e) es T cuando e < 0

**IF**

Tiene como argumentos: Una condición. Una o dos expresiones. Si la condición es T, se evalúa la primera condición, en caso contrario se evalúa la segunda.

12)

Planificación en IA:

* Numérica
* Lógica: Por pila de objetivos, No lineal, Jerárquica
* Reactiva

Planificación en robótica

* Reactiva
* De trayectoria de manipuladores

Planificación industrial: métodos basados en simulación discreta