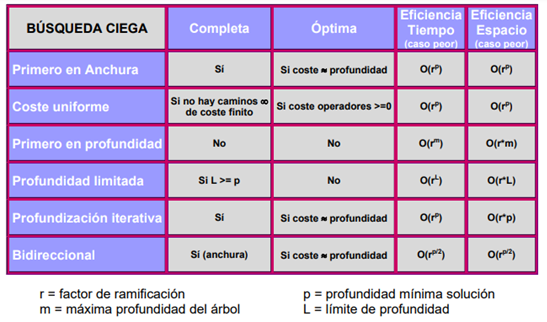
**EXAMEN TEMA 2**

Para la **búsqueda ciega** y la **búsqueda heurística**:

* Estado inicial
* Estado Final
* Definir acciones
* Indicar el factor de ramificación (número de estados que se generan por cada hijo) y profundidad
* Definir costes según acciones
* Indicar como es el espacio de estados (Expandir el árbol)
* Pensar una heurística (suele ser calcular la suma de elementos colocados/descolocados)
* Comprobar si es admisible -> si la heurística disminuye desde el inicial hasta el final suele estar bien
* Indicar un algoritmo adecuado para resolverlo.
* El algoritmo mejor es A\* porque es optimo completo y eficiente pero hay que decir otros
* 
* Completo -> Nos garantiza llegar al nivel donde se encuentra la solución (si la hay)

En caso de que se pida especificar **Algoritmo genético** hay que detallar:

* Tamaño y población inicial (Se genera de forma aleatoria) (TAM = 50 - 100)
* Función de fitness (si es de maximización o de minimización)
* Estrategia de selección
  + Torneo con presión de selección: se hacen torneos de p elementos y se enfrentan ganando en cada enfrentamiento el que mayor fitness tiene

Tipos: Determinista-> siempre gana uno

Probabilista-> fijar un valor (j) y de cada tirada seleccionar (a) número entre [0, 1]   
si a > j-> más apto  
si a < j -> menos apto

* + Ruleta (recta de marcaje)
* Probabilidad de cruce (0.8) y mutación (0.1 o 0.2)
* Estrategia destructiva o no destructiva para el cruce
* Función de cruce
  + Permutaciones (no repeticiones)
    - Cruce en orden (se copia la zona y desde la segunda marca se copia en orden de aparición)
    - Cruce PMX (se copia la zona y desde la primera marca se copia rebotando)
  + No permutaciones (repeticiones)
    - Cruce de 1 punto
    - Cruce de 2 puntos

En caso de ser una **Algoritmo de enfriamiento simulado**:

* Estado inicial generado aleatoriamente o no
* Cómo se generan los vecinos (función de mutación como el genético (mediante permutaciones))
* Función de valoración (Fitness) -> influye en la probabilidad de aceptación (depende de la función de calidad y de la temperatura)
* Temp inicial, temp final y velocidad de enfriamiento
  + Establecer una política de enfriamiento (sencillo como restar 1 o 2)
* Criterio de Parada

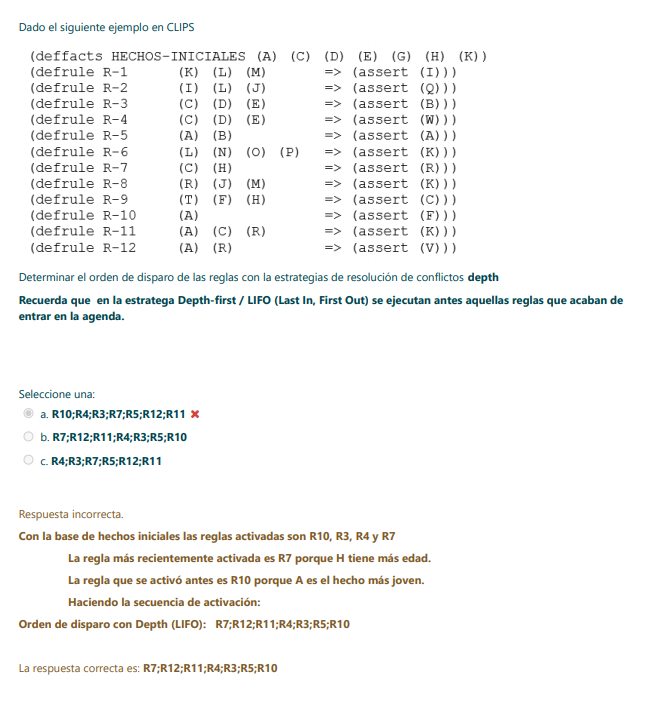
**Tema 3:**

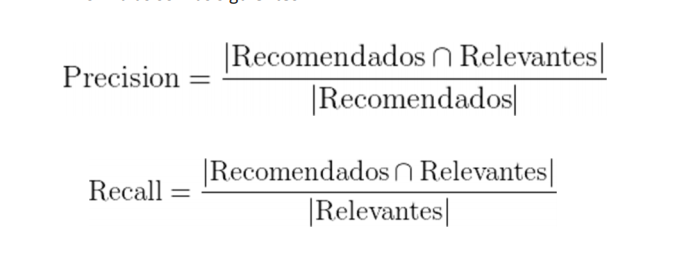
Ejercicio reglas:

Fijarse bien en la estrategia que usa ya que eso determina el orden de las reglas

En los valores de SPARQL:

buscamos en un saco con una relación y valor en concreto(OBJETO RELACIÓN VALOR)



tema 3: sistema basado en conocimiento(CBR):

* Recall -> del total de relevantes cuantos se han mostrado hasta la K
* Precisión -> del total (k) cuántos son relevantes. k -> índice(número que llevamos recorrido)

Clusters:buscamos las cantidades más pequeñas para ir juntando.

centroide-> usamos la media de los 2 extremos

simple los extremos más cercanos

completo->extremos

Codo:

Silhouette:

Matrices de confusión:

Exhaustividad (Recall): dividir el valor central por la suma de la fila->

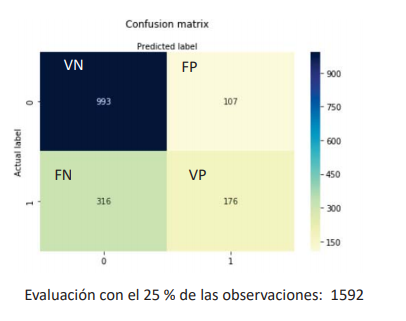
Precisión: Valor central entre la suma de la columna.-> falsos positivos

Global(exactitud): Aciertos / Total

la suma de la diagonal principal /total.

Recall: De todas las veces que es positivo cuántas veces ha acertado

Especificidad: De todas las veces que es negativo cuántas veces ha acertado



PARTE MATEO TEMA 2

Como hemos visto en clase la representación de un problema de espacio de estados consiste en:

-Representar estados y acciones mediante una estructura de datos.

-Definir: estado\_inicial, es\_estado\_final(), acciones(), aplica(,) y coste\_de\_aplicar\_accion, si el problema tiene coste.

HEURÍSTICAS:

-Heurística admisible si el coste real es mayor o igual que la heurística.

-Las heurísticas admisibles son optimistas si estiman que

el coste de solucionar el problema es menor que el que

realmente es.

-Una función heurística h1 es más informada que h2 si ambas son

admisibles y además para cualquier nodo n h(n) >= h1(n)>=h2(n)

Se dice que una función heurística es consistente sí solo sí para

todos los ni y nj (siendo nj descendiente de ni) se cumple:

h’(ni) – h’ (nj) <= coste(ni,nj); coste(ni,nj): coste para ir de ni a nj

A\*

- Para cada nodo

• Exploro sus hijos

\* El orden de exploración viene dado por el coste total previsto

(heurística + coste acumulado)

\* Primero exploro el hijo de coste total previsto menor

• Si el hijo es la primera vez que aparece

\* Calculo el coste total previsto y guardo el camino (añado el nodo

a la lista de abiertos)

• Si el hijo ya había aparecido antes y encuentro un camino

mejor que el que tenía guardado

\* Guardo el mejor y deshecho el otro

Solo se consigue la solución óptima si h es admisible.