# CAPÍTULO 1

## ||| Problemas de "algún camino" y de "el mejor camino" Ejemplos - 17/12/2015, 11/02/2016, 25/02/2016

**Algún camino:** Se resuelven mediante técnicas heurísticas que sugieran rutas adecuadas para explorar y se resuelven en tiempos razonables.

**Mejor camino:** Son más complicados de calcular y llevan mucho tiempo, pero se obtiene la mejor solución.

## | Definición de Espacio de Búsqueda 21/04/2014

Son los posibles lugares que se pueden recorrer entre el estado inicial y el final, incluyendo estos últimos.

## ||| Explicación de problemas recuperables, no recuperables e ignorables. ¿Cual requiere más planificación? - 17/12/2015, 31/07/2014, 25/02/2016

**Recuperables:** Se pueden deshacer pasos dados. Utilizan una estructura de control que se implementa como una pila Push-Down donde se almacenan las posiciones en caso que se necesiten deshacerse más tarde. Ejemplo: 8-Puzzle.

**No recuperables:** No se pueden deshacer pasos dados. Utilizan una estructura de control enfocada a la toma de decisiones. Ejemplo: Ajedrez.

**Ignorables:** Se pueden ignorar pasos dados. Ejemplo: Demostración de problemas.

Los problemas No Recuperables requieren más planificación ya que es necesario elegir correctamente cada paso porque no se puede volver atrás, para esto se analiza por adelantado una secuencia de pasos para descubrir la mejor opción a donde ir.

## || Problemas con consecuencia cierta e incierta - 06/08/2015, 31/07/2014

**Consecuencia cierta:** A través de una planificación puede conocerse exactamente qué ocurrirá. Ejemplo: 8-Puzzle.

**Consecuencia incierta:** No es posible una planificación con certeza. Suele tener una interacción externa. La planificación aplicada en este tipo de problemas suele ser muy cara, ya que el número de rutas que necesita explorar crece exponencialmente con el número de puntos en los cuales la consecuencia no puede predecirse.

Al combinar todos estos tipos de problemas, los de consecuencia incierta no recuperables son los más difícil de resolver. Ejemplo: Programación de un brazo robot.

## || Análisis de problemas - 03/12/2015, 11/02/2016

A fin de poder elegir el método más apropiado o una combinación de métodos para un problema en particular, es necesario analizarlo:

* ¿Puede **descomponerse** el problema en sub problemas independientes?  
  Por ejemplo, una integral que se pueda resolver por partes.
* ¿Pueden **deshacerse o ignorarse** pasos?  
  Problemas Ignorables, Recuperables y No recuperables.
* ¿Es **predecible** el universo del problema?

Consecuencia cierta e incierta

* ¿Una solución adecuada **es absoluta o relativa**?

¿Quiero llegar a la mejor solución o sólo encontrar una solución?

* ¿La solución es un **estado o una ruta**?

Solución estado: la solución es un estado. Solución camino: es la ruta hacia un estado objetivo.

* ¿Cuál es el papel del **conocimiento**? ¿es necesaria una gran cantidad de conocimiento para resolver el problema o solo es necesario para restringir la búsqueda?

Existen problemas en los que mucho conocimiento es necesario para acotar búsquedas y aquellos en los que el conocimiento se emplea para poder reconocer una solución.

* ¿Se necesita la **interacción** con una persona?

Solitarios o conversacionales.

# Preg. frecuentes adicionales del primer capitulo que recopile despues del 2016 hasta julio 2019:

## QUE ES IA:

Inteligencia Artificial es el estudio del comportamiento inteligente en las maquinas. A su vez el comportamiento inteligente supone percibir, aprender, razonar, comunicarse y actuar en entornos complejos. Las metas que tiene son:

-A largo plazo es el desarrollo de maquinas que puedan hacer las tareas que hacen los humanos.

-Otra meta es llegar a comprender este tipo de comportamiento.

## Aproximaciones a la IA:

Los principales paradigmas puede ser clasificados en dos grupos:

-Basadas en procesamientos de simbolos: son aproximaciones basadas en el conocimiento, en donde se aplican operaciones lógicas sobre bases de conocimiento declarativo. Éstas representan el conocimiento sobre un problema del dominio mediante sentencias declarativas, a menudo basadas en sentencias de la lógica de predicados.

-Aproximaciones subsimbolicas: Las redes neuronales son un ejemplo de maquinas que provienen de la escuela subsimbolica. Otras aproximaciones ascendentes, cercanas a las aproximaciones del tipo “vida artificial”, se basan en la Teoria de Control y en el Analisis de Sistemas Dinamicos.

## Tecnica IA:

Estas tecnicas existen para solucionar los problemas abordados por la IA. Una tecnica de ia es un metodo que utiliza conocimiento representado de tal forma que:

-el conocimiento representa generalizaciones.(se agrupan situaciones que comparten propiedades importantes).

-debe ser comprendido por las personas que lo proporcionan.

-puede modificarse facilmente para corregir errores.

-puede usarse en muchas situaciones a pesar de no ser preciso o completo.

Como conclusion se manifiestan 3 importantes tecnicas de IA:

Busqueda.

Uso del conocimiento.

Abstraccion.

## Descripcion formal de un problema

-definir un espacio de estados que contengan todas operaciones posibles

-definir los estados iniciales

-definir los estados objetivos

-definir conjunto de reglas que describan las acciones disponibles que se pueden realizar.

## Sistema de Produccion

Como es necesario estructurar los programas de IA para que se facilite la busqueda, los sistemas de produccion proporcionan tales estructuras.

Estos consisten en:

-Un conjunto de reglas compuestas por una parte izquierda(patron) que determina la aplicabilidad de la regla y una parte derecha que describe la operacion que se lleva a cabo si esta se aplica.

-Una o mas bases de datos/conocimientos

-Una estrategia de control que especifique el orden en que las reglas se comparan con la base de datos , para que el proceso de busqueda sea lo mas eficiente. Una buena est. de control debe generar algun cambio y debe ser sistematica.

-un aplicador de reglas.

## Caracteristicas de los sist. de Produccion ¿ Que relaciones existen entre los tipos de problemas y los tiopos de sist de produccion que son adecuados para resolver estos problemas?

Un sistema de produccion Monotono es aquel que la aplicacion de una regla nunca prevee la aplicacion de otra regla que podria haberse aplicado cuando se selecciono la primera. Uno no monotono es lo contrario al anterior.

Un sistema de produccion parcialmente conmutativo es aquel que tiene la propiedad de que si una determinada secuencia de reglas transforma el estado x en el y, entonces una permutacion de esta secuencia tambien cambia el estado x en y. Un sistema conmutativo es uno que es a la vez monotono y parcialmente conmutativo (Demostracion de teoremas).

# CAPÍTULO 2

## |||||| Técnicas de Búsqueda a Ciegas, Técnicas de Búsqueda Heurística. Diferencias 13/02/2014, 21/04/2014, 05/12/2013, 20/12/2012, 11/02/2016, 25/02/2016

Búsqueda: se busca una ruta a través del espacio, que una un estado inicial con un estado objetivo.

Técnica de Búsqueda a Ciegas: no se usa el conocimiento para llegar al objetivo.

Técnica de Búsqueda Heurística: es una técnica que aumenta la eficiencia de la búsqueda.

|  |  |  |
| --- | --- | --- |
|  | Ciegas | Heurística |
| Ventajas | Se obtiene la mejor solución. | Aumenta la eficiencia reduciendo el uso de recursos.  Se utiliza para resolver problemas complicados. |
| Desventajas | Demanda muchos recursos ya que se tienen que evaluar todas las opciones posibles.  Muchas veces no pueden resolverse problemas muy complicados | Se obtienen buenas soluciones, pero no se garantiza la mejor solución. |

## ||||| Búsqueda 1º en profundidad, 1º en anchura y 1º mejor. Ramificación y acotación. Ventajas, desventajas y diferencias. Algoritmo Mini-Max. - 31/07/2014, 27/02/2014, 21/04/2014, 13/02/2014, 05/12/2013, 22/11/2013

## ||| Escalada y los dos tipos de escalada - 17/12/2015, 03/12/2015, 13/02/2014

## ||| Funcionamiento de El primero Mejor - 17/12/2015, 11/02/2016, 25/02/2016

## ||| Verificación de restricciones, ¿cómo son los pasos? - 17/12/2015, 05/12/2013, 29/08/2013

## || Técnicas de búsqueda de juegos - 06/08/2015, 11/02/2016

## Técnicas de Búsqueda a Ciegas

### Explosión combinatoria

Se exploran todas las soluciones posibles. Es una búsqueda exhaustiva o “fuerza bruta”, que en teoría siempre funciona, pero en la práctica no, ya que consume muchos recursos.

### Ramificación y acotación

Inicialmente se generan rutas completas manteniéndose la ruta más corta encontrada hasta el momento. Deja de explorar una ruta tan pronto como su distancia total hasta el momento sea mayor que la que se ha marcado como la más corta. Se garantiza encontrar la ruta más corta, pero sigue consumiendo tiempo y recursos

### Primero en anchura

En este método, se parte de un estado inicial, se evalúa cada nodo del mismo nivel antes de proceder al siguiente nivel más profundo, hasta llegar a un estado objetivo. Se garantiza encontrar una solución, si existe. Eventualmente degenerará en una búsqueda exhaustiva.

### Primero en profundidad

En este método, se parte de un estado inicial, se explora cada camino posible hacia el objetivo hasta su conclusión, antes de intentar otro camino. Va por la izquierda hasta que se alcanza un nodo terminal, o se encuentra el objetivo. Si se alcanza un nodo terminal, se realiza una vuelta atrás hasta el nivel anterior y se sigue explorando de izquierda a derecha.

|  |  |  |
| --- | --- | --- |
|  | Anchura | Profundidad |
| Ventajas | No queda atrapada explorando callejones sin salida.  Se garantiza encontrar la solución más corta. | Necesita menos memoria porque almacena solamente los nodos del camino actual.  Puede encontrar una solución sin tener que examinar gran parte del espacio de estados. |

## Técnicas de Búsquedas heurísticas

### Generación y prueba

La técnica consiste en generar una posible solución utilizando una función heurística y verificar si la misma llega al objetivo. Para problemas más complicados no es muy eficiente.

### Escalada

Es una variante de Generación y Prueba. Existe realimentación a partir del procedimiento de prueba, que indica en qué dirección debe moverse. Se utiliza frecuentemente cuando se tiene una buena función heurística para evaluar los estados.

**Escalada simple:** Si el estado inicial no es objetivo, se genera un nuevo estado, se evalúa (mediante una función heurística) si es mejor que el actual, y si es mejor, el nuevo estado se convierte en actual. Se hace lo anterior hasta llegar a un estado objetivo.

**Escalada por máxima pendiente:** Es una variación el método anterior, con la diferencia de que antes de elegir el nuevo estado, considera todos los estados posibles a donde moverse y elige el mejor.

Problemas en las búsquedas de escalada:

**Máximo local**: cuando un estado actual es mejor que todos sus vecinos, pero no es mejor que algún estado en otros lugares. Solución: volver atrás hacia un nodo anterior e intentar seguir un camino diferente

**Meseta**: un conjunto de estados vecinos posee el mismo valor. Solución: realizar un gran salto en alguna dirección para intentar buscar en una nueva parte del espacio de búsqueda

**Cresta**: es un tipo especial de máximo local. Es un área del espacio de búsqueda más alta que las circundantes. Solución: aplicar dos o más reglas antes de realizar la evaluación.

### El primero mejor

Combina las ventajas de la búsqueda primero en anchura y la búsqueda primero en profundidad.

En cada paso, se selecciona el nodo más prometedor que se haya generado hasta ese momento, a través de una función heurística. A continuación, se expande el nodo elegido, aplicando las reglas para generar a sus sucesores. Si alguno de ellos es una solución, el proceso termina. Si no es así, estos nuevos nodos se añaden a la lista de nodos que se han generado hasta el momento. De nuevo, se selecciona el más prometedor de ellos y el proceso continúa de igual forma. La búsqueda puede volver a una rama una vez que los nodos de otras ramas sean lo suficientemente malos.

FALTA AGREGAR ALGORITMO, QUE LO PREGUNTÓ UNA VEZ

### Reducción de problemas

Consiste en convertir metas difíciles en una o más submetas más fáciles de lograr. Cada submeta, a su vez, puede dividirse más finamente en un o más submetas de nivel inferior.

La idea de este método es la exploración de un árbol de metas, donde los nodos representan metas y los hijos de cada nodo corresponden a sus metas.

* **Metas Y:** Se satisfacen solo cuando todas las submetas inmediatas quedan satisfechas.
* **Metas O:** Se satisfacen cuando cualesquiera de sus submetas quedan satisfechas.
* **Metas Hojas:** Se satisfacen directamente sin hacer referencia a ninguna submeta.

Los árboles de metas permiten responder preguntas sobre cómo y por qué se tomaron las acciones que se tomaron. La pregunta ¿cómo? es respondida a través de las submetas y la pregunta ¿Por qué? es respondida a través de la supermeta.

### Verificación de restricciones

El procedimiento consiste en propagar las restricciones tan lejos como sea posible. Entonces, si todavía no hay solución, se hace una suposición sobre algo y se añade como una nueva restricción. Entonces la propagación continúa con esta nueva restricción. Si en algún momento se encuentra una contradicción, puede usarse una vuelta atrás para reintentar con una suposición diferente.

La propagación puede terminar por: hallarse una contradicción, donde si dicha contradicción corresponde a las restricciones planteadas en la especificación del problema, no existe solución; o porque no puedan hacerse más cambios en base al conocimiento actual que se posea.

Por último, existen dos tipos de restricciones: posibles valores para un objeto y relaciones entre objetos.

### Análisis de medios y fines

Está centrado en la detección de diferencias entre el estado actual y el estado objetivo.

Una vez que se encontró una diferencia, debe encontrarse un operador que pueda reducirla. Es posible que tal operador no pueda ser aplicado en el estado actual, por lo que se crea el subproblema que consiste en alcanzar un estado en que pueda aplicarse dicho operador. Esto recibe el nombre de realización de subobjetivos para el operador.

Sin embargo, es posible que el operador no produzca el estado objetivo que se desea. En ese caso, se tiene un segundo subproblema que consiste en llegar desde ese estado hasta un objetivo.

Las reglas, que transforman un estado en otro, son representadas mediante un lado izquierdo, que contiene las precondiciones de la regla, y un lado derecho, que describe los aspectos del problema que cambiarán una vez aplicada la regla. Existe además una estructura llamada tabla de diferencias que ordena las reglas según las diferencias que pueden reducir.

## Búsqueda en problemas de juegos

Existen dos razones para que los juegos parezcan un buen dominio de exploración de la inteligencia en una máquina:

* Proporcionan una tarea estructurada en la que es muy fácil medir el éxito o el fracaso.
* No necesitan grandes cantidades de conocimiento, aunque esto no es cierto para juegos más complejos.

Para mejorar la efectividad de un programa resolutor de problemas es necesario:

* **Mejorar el procedimiento de generación**, de forma que sólo se generen buenos estados.
* **Mejorar el procedimiento de prueba**, de forma que sólo se exploren en primer lugar los buenos estados.

### Minimax

Es una búsqueda en profundidad limitada.

La idea consiste en comenzar en la posición actual y usar el generador de movimientos posibles para generar un conjunto de posiciones sucesivas. Entonces se puede aplicar la función de evaluación a esas posiciones y elegir simplemente la mejor. Después de hacer esto, puede llevarse hacia atrás ese valor hasta la posición de partida para representar nuestra evaluación de la misma. La posición de partida es exactamente tan buena para nosotros como la posición generada por el mejor movimiento que podamos hacer a continuación.

La alternancia de maximización y minimización en capas alternas cuando las evaluaciones se envían de regreso a la raíz, se corresponden con las estrategias opuestas que siguen los dos jugadores, y que da el nombre de Minimax.

### Alfa-Beta

Es una variante del Minimax que mejora su eficiencia usando técnicas de ramificación y acotación.

Se incluye una acotación para cada jugador, es decir, se requiere el mantenimiento de dos valores umbral, que representan la cota inferior llamada Alfa y la cota superior llamada Beta.

La forma en que se usan estas cotas consiste en:

La búsqueda en un nivel **minimizante** puede terminar cuando se descubre un valor menor que Alfa.

La búsqueda en un nivel **maximizante** puede terminar al descubrir un valor mayor que Beta.

## Componentes de un sistema de planificación

En los sistemas de resolución de problemas basados en técnicas elementales, era necesario llevar a cabo las siguientes funciones:

-Elegir la mejor regla para aplicar a continuación, basándonos en la mejor información heurística

disponible.

-Aplicar la regla elegida para calcular el nuevo estado del problema que surge de su aplicación.

-Detectar cuándo se ha llegado a una solución. Un sistema de planificación tiene éxito al encontrar

una solución cuando encuentra una secuencia de operadores que transforman el estado inicial del

problema en un estado objetivo.

-Detectar callejones sin salida, es decir, caminos que nunca pueden conducir a una solución, de

forma de que puedan abandonarse.

-Además, suele ser importante una quinta función:

Detectar cuándo se ha encontrado algo muy parecido a una solución correcta, y emplear técnicas

especiales para hacer que sea totalmente correcta.

## Estrategias de planificacion (son 4) / tipos de planificacion:

### Planificacion mediante una pila de objetivos:

En este método, el resolutor de problemas usa una pila que contiene tanto objetivos como operadores que deben proponerse para satisfacer estos objetivos. Este tipo de planificación aborda los problemas como objetivos conjuntos, resolviendo por orden los objetivos, uno cada vez. Genera un plan que contiene una secuencia de operadores que resuelven el primer objetivo, seguido por una secuencia completa para el segundo objetivo, y así.

### Planificación no lineal mediante fijación de restricciones

Los problemas difíciles provocan interacciones entre los objetivos. Los operadores que se utilizan para resolver un subproblema pueden interferir en la solución de un subproblema anterior. Así, se necesita un plan entrelazado en el que se trabaje simultáneamente con múltiples subproblemas. Este tipo de plan se denomina plan no lineal.

### Planificación jerárquica

Para problemas complejos, los resolutores de problemas tienen que generar planes muy extensos. Para poder hacerlo eficientemente, es importante poder eliminar algunos de los detalles del problema hasta que se encuentre una solución que resuelva los principales obstáculos. La asignación de valores de criticidad apropiados es un aspecto importante para el método de planificación jerárquica. Aquellas precondiciones que no tengan operadores que puedan satisfacerlas son las más críticas.

### sistemas reactivos

La idea de los sistemas reactivos consiste en evitar planificar totalmente y, en lugar de ello, utilizar la situación observable como pista a la que simplemente reaccionar. Un sistema reactivo debe tener acceso a alguna base de conocimiento que describa las acciones que deben realizarse bajo ciertas circunstancias.

cAPÍTULO 3

## ||||| Definición de argumento - 17/12/2015, 31/07/2014, 21/04/2014, 27/02/2014, 11/02/2016

Es un grupo de proposiciones o enunciados, que se dividen en premisas y conclusiones. Las premisas son fundamentos de las conclusiones.

## ||||| Estructura de un argumento - 17/12/2015, 06/08/2015, 27/02/2014, 20/12/2012, 11/02/2016

La **conclusión** de un argumento es la proposición afirmada basándose en las otras proposiciones del argumento y estas otras proposiciones que se afirman como fundamento o razones para la aceptación de la conclusión son las **premisas** de ese argumento.

## |||| ¿Cuándo un argumento es válido? - 17/12/2015, 31/07/2014, 21/04/2014, 11/02/2016

Un argumento es válido cuando sus premisas y conclusiones están relacionadas de modo tal que es absolutamente imposible que las premisas sean verdaderas, a menos que la conclusión lo sea también, es decir, tanto las premisas como las conclusiones tienen que ser verdaderas.

## || ¿Qué herramientas hay para probar esa validez? - 31/07/2014, 25/02/2016

Cuando las variables son pocas, podemos utilizar la tabla de verdad. Si las variables son muchas, podemos utilizar la Prueba Formal de Validez (mediante Reglas de Inferencia – 9 reglas) y Reglas de Reemplazo (mediante equivalencias – 10 equivalencias).

## ||||| ¿Qué es una tabla de verdad? Dar un ejemplo - 17/12/2015, 21/04/2014, 29/08/2013, 22/11/2013, 11/02/2016

Es una representación de los valores de verdad de un enunciado, que están determinados a partir de los valores de verdad de sus componentes.

## ||| Tablas de verdad. ¿Para qué sirven y cómo se construyen? - 06/08/2015, 13/02/2014, 20/12/2012

Las tablas de verdad sirven para demostrar la validez de un argumento y se construyen ubicando en sus columnas los enunciados del argumento (las premisas y la conclusión) y en sus filas todas las combinaciones posibles de los estados.

## |||| Si tengo una tabla completa, como me doy cuenta si el argumento es válido - 17/12/2015, 22/11/2013, 11/02/2016, 25/02/2016

## Para que un argumento sea válido, hay que corroborar que en las filas (instancias de sustitución), donde la conclusión es verdadera, las premisas deben ser verdaderas. Además, siempre que la conclusión sea falsa, alguna de las premisas debe ser falsa.

## ||||| Qué es una prueba formal de validez - 17/12/2015, 06/08/2015, 13/02/2014, 20/12/2012, 25/02/2016

Una prueba formal de validez es la demostración de que un argumento es válido a partir de las 9 reglas de inferencia.

## ||| Los 2 tipos de demostración de invalidez con explicación (regla de demostración condicional y regla de demostración indirecta) - 17/12/2015, 03/12/2015, 06/08/2015

**Regla de demostración condicional:** Tener en cuenta que en la conclusión tiene que haber un condicional, sino no se puede aplicar. Se construye suponiendo que el antecedente de su conclusión es una **premisa** adicional y luego deduciendo el **consecuente** de su conclusión por una sucesión de argumentos válidos elementales.

**Regla de demostración indirecta (absurdo):** Se parte suponiendo lo opuesto de lo que se quiere demostrar, si esto conduce a una contradicción, entonces el supuesto es falso y su negación es verdadera.

## |||| Conocimiento declarativo y conocimiento procedimental - 17/12/2015, 21/04/2014, 27/02/2014

**Representación declarativa:** El conocimiento está especificado, pero no la manera de usarlo. Para utilizar este tipo de representación, es necesario un programa que especifique lo que debe hacerse con ese conocimiento y de qué forma.

**Representación procedimental:** La información de control para utilizar el conocimiento, se encuentra dentro del propio conocimiento. Para utilizar esta representación, se necesita un intérprete que siga las instrucciones dadas por el conocimiento.

## | Operadores de relación entre enunciados - 17/12/2015

Los enunciados se dividen en simples y compuestos.

Un enunciado simple es el que no contiene otro enunciado como parte componente.

Un enunciado compuesto contiene otro enunciado como componente.

1. Conjunción: Es equivalente a un AND. Se representa con “.”
2. Negación: Es equivalente a lo opuesto. Se representa con “~”
3. Disyunción: Es equivalente a un OR. Se representa con “v”
4. Condicional: Es equivalente a ~(p.~q). se representa con “⊃”

## || Qué son reglas de inferencia y reglas de reemplazo. ¿Cómo se usan? Diferencias - 17/12/2015, 06/08/2015

Las **reglas de inferencia** son nueve formas de argumento válidas elementales que pueden usarse al construir pruebas formales de validez.

Hay muchos argumentos válidos cuya validez no se puede probar usando solamente las nueve reglas de inferencia, para estos casos se reemplaza una parte de un enunciado compuesto por una expresión que es lógicamente equivalente. A esto se le llama **reglas de reemplazo**. Existen diez expresiones lógicamente equivalentes.

## ||| Equivalencia lógica - 06/08/2015, 11/02/2016, 25/02/2016

**Tautología**: Se da cuando los argumentos tienen el mismo valor de verdad (el mismo valor en las conclusiones, en todas las instancias de sustitución).

**Equivalencia Lógica o ser Materialmente Equivalentes**: Tener una tautología implica que existe una equivalencia lógica o que los argumentos son materialmente equivalentes.

## || ¿Cuándo no se pueden usar tablas de verdad y qué se utiliza en este caso? - 13/02/2014, 11/02/2016

Cuando las variables son muchas no se pueden utilizar las tablas de verdad. En este caso se utiliza la prueba formal de validez.

## || Diferencia entre lógica de predicados y lógica proposicional - 05/12/2013, 29/08/2013

La **lógica proposicional** (o lógica de orden cero), como medio de representación del conocimiento, es atractiva debido a su simplicidad.

Sin embargo, hay cosas que no son representables utilizando lógica proposicional, por lo que es necesario utilizar la **lógica de predicados** o de primer orden, en donde aparecen los cuantificadores para representar el mismo.

**La programación lógica (la que utiliza Prolog) está basada en la lógica de predicados.**

## | ¿Cómo se llega a la lógica de predicados? - 27/02/2014

## || Representación del conocimiento bajo incertidumbre. ¿Por qué es necesaria?

En los enfoques anteriores de representación del conocimiento nos basamos en un modelo donde la información es completa, consistente e inalterable. La representación del conocimiento bajo incertidumbre es necesaria ya que en muchos problemas no es posible crear tales modelos.

## |||| Técnicas de razonamiento bajo incertidumbre - 05/12/2013, 29/08/2013, 22/11/2013

**Razonamiento no monótono**: Posibilitan la obtención de suposiciones cuando no existen evidencias que las contradigan, es decir, se puede suponer una hipótesis nueva siempre y cuando no hay una hipótesis que diga lo contrario.

**Razonamiento estadístico**: Se agrega algún tipo de medida numérica sobre la certeza de la suposición.

FALTA COMPLETAR CON LOS TIPOS DE CADA UNO.

## | Factores de certeza - 05/12/2013

En una base de conocimiento, donde éste no es completo, a cada regla se le asocia un factor de certeza, que representa una medida sobre la evidencia que existe sobre la conclusión:

Un factor de certeza se define en términos de dos componentes:

MB[h, e]: una medida entre 0 y 1 de la creencia (belief) de que la hipótesis h proporciona la evidencia e.

MD[h, e]: una medida entre 0 y 1 sobre la incredulidad (disbelief) de que la hipótesis h proporciona la evidencia e.

A partir de estas dos medidas, se define el factor de certeza como:

FC[h, e] = MB[h, e] – MD[h, e]

## | Estructuras de relleno fuertes y débiles - 05/12/2013

Estructuras de ranura y relleno **débiles**: fueron introducidas para soportar adecuadamente la herencia. En estas estructuras hablamos de “conocimiento pobre”.

* **Redes semánticas**: La información se representa como un conjunto de nodos conectados mediante un conjunto de arcos etiquetados que representan las relaciones entre ellos.
* **Marcos**: Es una colección de atributos con valores asociados, que describe alguna entidad del mundo. Cada marco puede representar una clase o una instancia.

Estructuras de ranura y relleno **fuertes**: las herramientas utilizadas en este caso implementan poderosas teorías sobre la forma de representar y utilizar el conocimiento.

* **Dependencia conceptual**: Es una forma de representar el conocimiento que tiene como objetivo facilitar extraer inferencias de las frases y ser independiente del lenguaje original de las mismas.
* **Guiones**: Describe una secuencia de eventos (a diferencia de la dependencia conceptual que describe eventos por separado)
* **CYC**: Es un proyecto de una gran base de conocimiento cuyo propósito es de capturar el conocimiento humano de sentido común.

## Metodo de resolucion para que sirve(estaba en la logica proposicional):

## Algoritmo de unificación:

(no recuerdo si estaba en el capitulo 2 o 3)

## Que dos condiciones debe cumplir un argumento para que que la conclusión sea verdadera:

El argumento debe ser válido y todas sus premisas deben ser verdaderas.

## Que garantiza la falsedad de su conclusión

que todas sus premisas sean falsas.

## Armar la tabla de verdad y justificar la validez o invalidez de:

"Si el catálogo de semillas es correcto, entonces si las semillas se siembran en abril, entonces si las flores se abren en julio. Las flores no se abren en julio. Por lo tanto si las semillas se siembran en abril, entonces el catálogo no es correcto".

## -Cómo se compone un argumento, ¿Es inductivo o deductivo?

se compone por las premisas y la conclusión. Existen argumentos inductivos y deductivos pero nosotros nos enfocamos en el libro en los deductivos ya que La **conclusión** de un argumento es la proposición afirmada basándose en las otras proposiciones del argumento y estas otras proposiciones que se afirman como fundamento o razones para la aceptación de la conclusión son las **premisas** de ese argumento.

## - Si tenes un argumento y su conclusión es falsa, qué puedes decir del argumento?

Nada, tenés que ver las premisas para afirmar o negar la validez; ver si son todas V o hay alguna F.

El argumento será válido si existe una instancia de sustitución donde todas las premisas sean verdaderas y también su conclusión, y **además** si en una instancia de sustitución la conclusión es falsa, algunas de todas sus premisas debe ser verdaderas

## -Refutación por analogía lógica:

si la forma específica de un argumento dado puede mostrarse que tiene una instancia de sustituir con premisas verdaderas y conclusión falsa, entonces el argumento dado es invalido.

## -forma sentencial

es cualquier sucesion de simbolos conteniendo variables sentenciales. De modo que al sustituir enunciados por variables sentenciales el resultado es un enuncidoos

# CAPÍTULO 4

## ||||| Ventajas de un sistema experto - 17/12/2015, 27/02/2014, 29/08/2013, 20/12/2012, 11/02/2016

1. Está accesible para usarse en **cualquier momento** (a diferencia de un experto humano que tiene que comer, dormir, etc).
2. Pueden crearse **infinitos sistemas expertos**, mientras que el número de expertos humanos es limitado.
3. Dado que se puede **copiar fácilmente** el conocimiento, se dice que los sistemas expertos nunca mueren.
4. Su **rendimiento siempre es el máximo** (no está influenciado, por ejemplo, por el cansancio).
5. Al **no tener personalidad**, no tienen problemas a la hora de expresar su conocimiento.
6. Se puede adquirir un nuevo sistema experto copiando el programa de una máquina a otra.

## |||||||| Estructura de un sistema experto - 17/12/2015, 03/12/2015, 06/08/2015, 31/07/2014, 27/02/2014, 22/11/2013, 21/04/2014, 25/02/2016

En un sistema experto se independiza el conocimiento de los procedimientos que hacen uso de él, por tanto, obtenemos dos módulos bien diferenciados: la base de conocimiento y el motor de inferencia.

Podemos distinguir tres componentes estructurales básicos:

1. **Base de hechos**: contiene el conocimiento declarativo, a nivel de datos, sobre el problema y el estado del sistema en un momento dado.
2. **Base de conocimiento**: contiene el conocimiento específico y procedimental acerca de la clase de problemas en los que el sistema es experto.
3. **Motor de inferencia**: contiene las funciones deductivas del sistema.

Tipos de personas que se relacionan con un sistema experto:

1. El **usuario final**, que dialoga con el sistema para resolver problemas o aprender.
2. El **experto humano**, que comunica su conocimiento para construir el sistema.
3. El **ingeniero de conocimiento**, que diseña las estructuras de datos para la representación del conocimiento y traduce el conocimiento humano al sistema.

## ||| ¿Qué razonamiento usa Prolog y qué tipo de búsqueda? - 06/08/2015, 31/07/2014, 13/02/2014

El **razonamiento** que utiliza Prolog es **“hacia atrás”**, partiendo de estados objetivos. Se comienza desde el objetivo, se van generando niveles hasta que se genere un nodo que se empareja con el estado inicial.

El tipo de búsqueda que utiliza Prolog **es primero en profundidad con vuelta atrás**.

## ||| Definición funcional de un sistema experto - 21/04/2014, 27/02/2014, 13/02/2014

Para que un sistema pueda llamarse “experto” tiene que satisfacer un criterio similar al Test de Turing, es decir, tiene que ser indistinguible de un experto humano.

Un sistema experto debe ser capaz de:

* **Resolver** problemas muy difíciles tan bien o mejor que un experto humano.
* **Razonar**.
* **Interactuar** en lenguaje humano.
* **Funcionar con datos erróneos** y reglas imprecisas.
* Contemplar **múltiples hipótesis**.
* **Justificar** sus conclusiones.

## Armazones o shells

En un principio, los sistemas expertos que se construían eran creados desde cero. Luego se vio que tenían varios puntos en común. En particular, desde que los sistemas se creaban como un conjunto de representaciones declarativas, acompañadas por un intérprete para dichas declaraciones, fue posible separar el conocimiento específico del intérprete. Los intérpretes resultantes se llaman armazones o shells.

Los primeros armazones de sistemas expertos proporcionaban mecanismos para representación del conocimiento, el razonamiento y la explicación. Más tarde, se les añadieron herramientas para la adquisición del conocimiento. Por último, se facilitó la integración con otros tipos de programas.

## Lenguaje procedural vs. lenguaje declarativo

Los lenguajes **procedurales** permiten al programador decirle a la computadora lo que tiene que hacer, paso a paso, procedimiento por procedimiento, hasta alcanzar una conclusión.

Los lenguajes **declarativos**, como PROLOG, necesitan que se declaren reglas y hechos sobre símbolos específicos, y luego se le pregunte sobre si un objetivo concreto se deduce lógicamente a partir de los mismos. El propio lenguaje realiza el trabajo de decidir cómo alcanzar dicho objetivo.

# CAPÍTULO 5

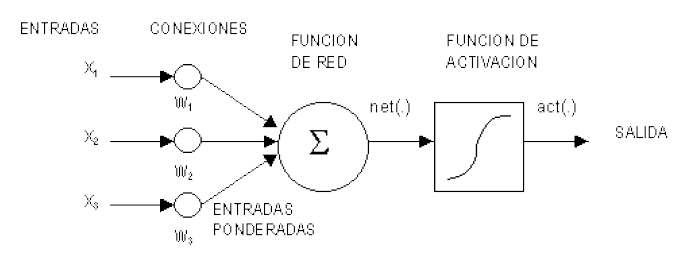
## | Redes neuronales. ¿Qué son? - 13/02/2014

Es un sistema compuesto de muchos elementos simples de procesamiento, los cuales operan en paralelo y cuya función es determinada por la estructura de la red y el peso de las conexiones.

## || Estructura de una red neuronal. Explicar función de red o de propagación y función de activación – 11/02/2016, 25/02/2016

Las neuronas son modeladas mediante **unidades de proceso**. Cada unidad de proceso se compone de:

* Una red de conexiones de **entrada**.
* Una **función de red**, o propagación, que calcula el valor de entrada total a la unidad, generalmente es la suma ponderada de todas las entradas recibidas.
* Un **núcleo** central **de proceso**, encargado de aplicar la función de activación, que se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total. Estas funciones pueden ser lineales, funciones de umbral o no lineales.
* Una **salida**, por donde se transmite el valor de activación a otras unidades.



## Comparación entre redes neuronales biológicas y artificiales

|  |  |
| --- | --- |
| **Redes neuronales biológicas** | **Redes neuronales artificiales** |
| Neuronas | Unidades de proceso |
| Conexiones sinápticas | Conexiones ponderadas |
| Efectividad de la sinapsis | Peso de las conexiones |
| Efecto excitatorio o inhibitorio de una conexión | Signo del peso de una conexión |
| Efecto combinado de la sinapsis | Función de red o propagación |
| Activación 🡪 tasa de disparo | Función de activación 🡪 salida |

## | Características de las redes neuronales artificiales - 21/04/2015

1. **Aprendizaje inductivo:** no se le indican reglas, modifica su comportamiento en función de la experiencia.
2. **Generalización:** una vez que está entrenada, se le pueden presentar datos distintos a los usados en el aprendizaje. La respuesta obtenida dependerá del parecido de los datos con los ejemplos del entrenamiento.
3. **Tolerancia al ruido:** pueden procesar datos incompletos o distorsionados.
4. **Procesamiento paralelo:** las redes neuronales están pensadas para el procesamiento en paralelo mediante multiprocesadores.
5. **Memoria distribuida:** el conocimiento acumulado de la red se haya distribuidos en numerosas conexiones. Esto tiene como consecuencia la tolerancia a fallos.

## ||||| Ventajas de las redes neuronales - 06/08/2015, 13/02/2014, 21/04/2014, 27/02/2014, 05/12/2013

Ventajas:

1. Aprendizaje adaptativo: aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos.
2. Auto organización: la auto organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico.
3. Tolerancia a fallos: la destrucción parcial de una red conduce a una degradación de su estructura, sin embargo, algunas capacidades de la red se pueden retener.
4. Operación en tiempo real: los cómputos neuronales pueden ser realizados en paralelo, para lo cual se diseña hardware especial.
5. Fácil inserción dentro de la tecnología existente: pueden obtenerse chips especializados para redes neuronales que mejoran su capacidad.

## ||||| Desventajas de las redes neuronales - 17/12/2015, 13/02/2014, 21/04/2014, 05/12/2013, 25/02/2016

Desventajas:

1. Definición de **muchos parámetros** antes de aplicar la metodología.
2. **Caja negra**. No ofrecen una fácil interpretación de cómo toman sus decisiones y por lo tanto no se puede explicar por qué motivos se tomó una decisión.

## ||| Métodos de aprendizaje de las redes neuronales - 17/12/2015, 13/02/2014, 25/02/2016

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Cuando los pesos se mantienen estables se dice que la red ha aprendido.

Métodos de aprendizaje:

1. Aprendizaje **supervisado**: es controlado por un agente externo (supervisor) que determina la respuesta que debería generar, a partir de una entrada determinada. En caso de que la salida no sea la esperada, se modifican los pesos de las conexiones.

Tipos de aprendizajes supervisados:

* 1. Aprendizaje por **corrección de error**: se ajustan los pesos de las conexiones con el objetivo de reducir el error cometido:

Error cometido = Valor deseado – Valor obtenido

* 1. Aprendizaje **por refuerzo**: es más lento que el anterior, debido a que el supervisor se limita a indicar si la salida es éxito o fracaso.
  2. Aprendizaje **estocástico**: consiste en realizar cambios aleatorios de los valores de los pesos. Luego de esto, se evalúa la energía de la red, y se acepta el cambio si ésta es menor que la energía anterior.

1. Aprendizaje **no supervisado**: no requiere influencia externa para ajustar los pesos de las conexiones. No recibe información del entorno.

Tipos de aprendizajes no supervisados:

* 1. **Aprendizaje hebbiano:** cuando dos células están próximas, el peso de la conexión entre ambas se va incrementando en un período de tiempo. Ejemplo del perro, la campanita y la comida.
  2. **Aprendizaje competitivo y comparativo:** se orienta a la clasificación de los datos de entrada, las unidades de salida luchan por el control sobre porciones del espacio de entrada.

## || Componentes de un perceptrón - 29/08/2013, 22/11/2013

Un **perceptrón** imita a una neurona tomando la suma ponderada de sus entradas y enviando a la salida un 1 si la suma es más grande que algún valor umbral ajustable, y enviando un 0 en caso contrario.

En un perceptrón:

1. Sólo hay una neurona.
2. Las entradas son binarias.
3. Las cajas lógicas pueden interponerse entre las entradas y los pesos del perceptrón.  
   La caja lógica produce un valor de salida de 0 o 1 para cada combinación de 0 y 1 que pueda aparecer en sus entradas.
4. La salida del perceptrón es 0 o 1 dependiendo de si la suma ponderada de las salidas de las cajas lógicas es mayor que el umbral.

## Aprendizaje del perceptrón

Existe un procedimiento que determina un buen conjunto de pesos para un perceptrón. Se empieza con todos los pesos en cero. Después se intenta el perceptrón con todas las muestras, una a la vez. Siempre que el perceptrón cometa un error, se cambian los pesos de modo que el error se haga menos probable; en cualquier otro caso, no se hace nada.

Para entrenar a un perceptrón:

Hasta que el perceptrón produzca el resultado correcto para cada muestra de entrenamiento:

* Si el perceptrón produce una respuesta errónea:
  + Si el perceptrón dice NO cuando debería decir SI, añada el vector de salida de caja lógica al vector peso.
  + En otro caso, reste el vector de salida de caja lógica al vector peso.
* En cualquier otro caso, no haga nada.

## | Redes de Hopfield. Redes recurrentes. Redes de Jordan - 20/12/2012

Pagina 24 a 29