



INVESTIGACION 3: MOTORES DE INFERENCIA DE LA ACTUALIDAD

Materia: Sistemas Expertos

Alumno: Eric Uriel Cabrera Hernández 22310206

Motores de inferencia

Hoy en día los motores de inferencia han evolucionado muchísimo. No se limitan ya a los clásicos sistemas expertos de los 80s–90s, sino que se usan en tres grandes áreas: sistemas basados en reglas, ontologías/semántica, y aprendizaje automático (IA moderna).

PRIMERO CONVIENE DISTINGUIR DOS GRANDES GRUPOS:

1. Motores de inferencia basados en **reglas / lógica simbólica** (producción de reglas, lógica, ontologías, RDF, etc.).
 - Los **motores de reglas/lógica** se centran en reglas explícitas SI–ENTONCES para dominios específicos.
 - Los **motores de semántica/ontologías** trabajan con estructuras de conocimiento jerárquicas y **relaciones semánticas**, lo que les permite inferir propiedades y significados más generales.
2. Motores de inferencia empleados en IA moderna / aprendizaje automático / “neuro-simbólico” que combinan reglas + redes neuronales.

EJEMPLOS DE MOTORES DE INFERNICIA MODERNOS BASADOS EN REGLAS / LÓGICA

- **Drools**

Es un motor de reglas de negocio en Java, muy usado para sistemas con muchas reglas, lógica empresarial, flujo de trabajo, etc. Soporta forward chaining y demás.

- **CLIPS / FuzzyCLIPS**

CLIPS es una “expert system shell” clásica (producción de reglas, etc.). FuzzyCLIPS extiende CLIPS para manejar lógica difusa (fuzzy logic).

- **JESS**

Similar a CLIPS, pero en Java; permite integración en aplicaciones Java. Es muy usado para reglas de decisión, etc.

- **Apache Jena**

Sistema para trabajar con RDF / ontologías / triples. Tiene “reasoners / inference support” para trabajar con ontologías y reglas. Puede usar diferentes modalidades de inferencia como forward chaining.

- **Stardog**

Es una plataforma de grafo de conocimiento (“knowledge graph”) con motor de inferencia para RDF/ontologías. Permite inferir relaciones adicionales basadas en esquemas, reglas declarativas, ontologías, etc.

- **CWM (Closed World Machine)**

Motor de inferencia forward chaining para Semantic Web, que trabaja con RDF/RDF-N3, reglas, etc.

- **OPS5**

Uno de los clásicos: motor de reglas basado en forward chaining, con algoritmo RETE. Se usa menos ya para nuevos proyectos, pero fue muy influyente.

Ejemplos de motores de inferencia en IA moderna / aprendizaje / modelos híbridos

- **NELLIE (Neuro-Symbolic)**

Es una propuesta reciente que combina razonamiento simbólico con componentes de aprendizaje / recuperación para construir explicaciones (“proof trees”). Intenta unir reglas / lógica con modelos de lenguaje natural.

- Motores para inferencia de modelos de deep learning: aunque no son “sistemas expertos” tradicionales, muchas aplicaciones usan motores que hacen inferencia eficiente de redes neuronales:

- **TensorRT** de NVIDIA
- **OpenVINO** de Intel
- **ONNX Runtime**
- **TensorFlow Lite** (para dispositivos móviles y embebidos)

Estos motores ayudan cuando tienes modelos entrenados y quieres ejecutar (“inferir”) con baja latencia, pero no todos implementan reglas explícitas: suelen ser redes neuronales, no reglas SI-THEN.

Cómo se elige uno / criterios importantes

Cuando alguien decide qué motor usar, suele fijarse en:

- ¿Qué tipo de conocimiento se representa? (reglas explícitas, ontologías, fuzzy logic, etc.)
- ¿Necesita explicación / trazabilidad del razonamiento?
- ¿Performance / latencia? ¿Hay datos en tiempo real?
- ¿Capacidad para manejar incertidumbre, excepciones?
- ¿Facilidad de integración con sistemas existentes?
- En IA: tamaño del modelo, hardware disponible, compatibilidad, eficiencia.

DIFERENCIA ENTRE MOTORES DE REGLAS/LÓGICA Y MOTORES DE SEMÁNTICA/ONTOLOGÍAS

Aspecto	Motores de reglas / lógica	Motores de semántica / ontologías
Forma de representar conocimiento	Reglas SI-ENTONCES (producción de reglas), lógica proposicional o lógica de predicados.	Ontologías en RDF, OWL, RDFS → definen conceptos, relaciones y jerarquías .

Aspecto	Motores de reglas / lógica	Motores de semántica / ontologías
Base de conocimiento	Conjunto de reglas y hechos explícitos.	Grafo semántico de conceptos conectados por relaciones.
Inferencia	Encadenamiento hacia adelante o hacia atrás (forward / backward chaining).	Razonamiento por herencia, subsunción, consistencia ontológica.
Aplicación típica	Diagnóstico médico, sistemas industriales, soporte a decisiones.	Web semántica, integración de datos, asistentes inteligentes, knowledge graphs.
Ventajas	<ul style="list-style-type: none"> - Explicables y fáciles de rastrear. - Adecuados para dominios concretos y bien definidos. 	<ul style="list-style-type: none"> - Permiten razonar sobre grandes volúmenes de datos enlazados. - Manejan jerarquías complejas y herencia de propiedades.
Limitaciones	<ul style="list-style-type: none"> - Escalan mal con miles de reglas. - Difícil manejar conceptos generales o ambigüedad. 	<ul style="list-style-type: none"> - Necesitan ontologías bien diseñadas (costo alto). - Más lentos para inferencias complejas.

Ejemplos de motores CLIPS, Jess, Drools, OPS5.

Apache Jena, Stardog, GraphDB, Pellet.

◆ Motor de reglas/lógica

- Regla: **SI** fiebre > 38°C **Y** tos seca → **ENTONCES** posible gripe.
- El motor compara hechos (síntomas) con reglas y da un diagnóstico.

◆ Motor de semántica/ontologías

- Ontología:
 - “La gripe es una enfermedad viral”.
 - “Toda enfermedad viral es contagiosa”.
- Inferencia: si algo es *gripe*, entonces es *contagiosa*, aunque no haya una regla **SI-ENTONCES** explícita.

Métodos de inferencia actuales

Encadenamiento hacia adelante (Forward chaining)

El encadenamiento hacia adelante es un método de inferencia que parte de los hechos iniciales conocidos y aplica reglas de la base de conocimientos para generar nuevos hechos hasta llegar a una conclusión o meta.

Características principales

- **Punto de partida:** hechos iniciales (información que se conoce desde el inicio o se recibe de sensores/usuario).
- **Proceso:** aplica reglas del tipo SI... ENTONCES para derivar nuevos hechos.
- **Finalización:** se detiene cuando:
 1. Ya no se pueden aplicar más reglas, o
 2. Se alcanza una conclusión/meta deseada.
- **Tipo de razonamiento:** progresivo, va construyendo conocimiento nuevo a medida que avanza.

Cómo funciona paso a paso

1. **Base de hechos inicial:** se cargan los hechos conocidos (ejemplo: "temperatura = 39°C").
2. **Selección de reglas aplicables:** el motor de inferencia revisa qué reglas tienen sus condiciones satisfechas con los hechos actuales.
 - Ejemplo: *Si temperatura > 38°C → ENTONCES fiebre.*
3. **Aplicación de la regla:** si la condición se cumple, la acción de la regla se ejecuta y genera un nuevo hecho.
 - Nuevo hecho: "El paciente tiene fiebre".
4. **Actualización de la base de hechos:** el nuevo hecho se añade a la memoria de trabajo (working memory).
5. **Repetición del ciclo:** se buscan nuevas reglas que puedan aplicarse con la información actualizada.
6. **Finalización:** cuando ya no quedan reglas aplicables o se alcanza el objetivo.

Ventajas

- Natural en sistemas donde la información se va capturando desde los datos (sensores, usuario, bases de datos).
- Puede descubrir múltiples conclusiones a partir de los mismos hechos.
- Es fácil de implementar con reglas de producción.

Desventajas

- Puede ser ineficiente si hay muchísimas reglas, ya que evalúa todas las posibles.
- Puede generar conclusiones innecesarias si no hay un objetivo definido claro.
- Consumo más recursos en sistemas grandes y complejos.

2. Encadenamiento hacia atrás (Backward chaining)

El encadenamiento hacia atrás es un método de inferencia que parte de una meta o hipótesis y trabaja hacia atrás, buscando qué hechos o condiciones deben cumplirse para confirmarla. Se dice que es un razonamiento guiado por metas (goal-driven).

Características principales

- **Punto de partida:** una hipótesis o meta a comprobar (ejemplo: "¿El paciente tiene gripe?").
- **Proceso:** se buscan reglas que concluyan la meta → se revisan sus premisas → si no están en la base de hechos, se intentan probar como nuevas metas.
- **Finalización:** se confirma o se rechaza la hipótesis.
- **Tipo de razonamiento:** regresivo, trabaja de la conclusión hacia los datos.

Cómo funciona paso a paso

1. **Meta inicial:** el sistema recibe una hipótesis (ej. "el paciente tiene gripe").
2. **Búsqueda de reglas:** el motor de inferencia busca en la base de conocimientos reglas cuyo consecuente coincida con la meta.
 - Ejemplo: *Si fiebre Y tos → ENTONCES gripe.*
3. **Planteamiento de sub-metas:** para confirmar la meta, ahora deben verificarse los hechos: "fiebre" y "tos".
4. **Verificación de hechos:**
 - Si los hechos ya están en la base de datos → se confirman.
 - Si no lo están → se convierten en nuevas metas y se buscan reglas que los justifiquen.
5. **Resolución recursiva:** se repite el proceso de búsqueda de reglas y sub-metas hasta que:
 - Se encuentra evidencia suficiente en la base de hechos, o
 - No se pueden probar las premisas (entonces la hipótesis es falsa).
6. **Conclusión:** si todas las sub-metas se cumplen → la hipótesis inicial se confirma.

Ventajas

- Muy eficiente cuando se tiene una meta específica.

- Evita explorar reglas innecesarias (a diferencia del forward chaining).
- Ideal para sistemas de diagnóstico (médico, técnico, legal, etc.).

Desventajas

- Puede ser ineficiente si hay muchas metas posibles y pocas reglas para llegar a ellas.
- Si los hechos no están disponibles y no se pueden deducir → no se logra concluir.
- Menos útil en sistemas donde la información fluye desde sensores o datos en tiempo real (ahí conviene más el forward chaining).

3. Razonamiento basado en reglas de producción

El razonamiento basado en reglas de producción es un enfoque de inferencia donde el conocimiento se representa como un conjunto de reglas del tipo SI-ENTONCES (IF-THEN), llamadas producciones, y se aplica un motor de inferencia para combinar estas reglas con hechos de la base de datos y así obtener conclusiones o acciones.

Estructura de una regla de producción

Cada regla tiene dos partes:

- **Condición (parte izquierda / antecedente):** describe una situación o conjunto de hechos.
 - Ejemplo: *Si la temperatura > 38°C y hay tos...*
- **Acción (parte derecha / consecuente):** indica qué hacer si la condición se cumple.
 - Ejemplo: *ENTONCES diagnosticar gripe.*

Cómo funciona el razonamiento por reglas de producción

1. **Base de conocimientos:** contiene todas las reglas de producción.
2. **Base de hechos:** contiene la información inicial (hechos conocidos).
3. **Motor de inferencia:**
 - Compara los hechos con las condiciones de las reglas.
 - Selecciona las reglas aplicables (conflicto).
 - Ejecuta las acciones de las reglas seleccionadas (resolución de conflicto).
 - Actualiza la base de hechos con nuevos datos.
 - Repite el ciclo hasta que no queden reglas aplicables o se alcance una meta.

Mecanismos clave en el razonamiento basado en reglas

- **Ciclo de reconocimiento-acción (recognize-act cycle):**
 - Reconocer hechos → identificar reglas aplicables → ejecutar acciones → actualizar hechos.

- **Resolución de conflictos:** cuando varias reglas son aplicables, se deben priorizar usando estrategias como:
 - Orden de las reglas.
 - Especificidad de las condiciones.
 - Recencia de los hechos.
- **Modo de encadenamiento:** el motor de inferencia puede aplicar:
 - Hacia adelante (data-driven).
 - Hacia atrás (goal-driven).

Ventajas

- Muy intuitivo y cercano al razonamiento humano.
- Flexibilidad: fácil de añadir, modificar o eliminar reglas.
- Modularidad: cada regla es independiente, lo que facilita el mantenimiento.
- Explicabilidad: el sistema puede mostrar qué reglas usó para llegar a una conclusión.

Desventajas

- Escalabilidad: con miles de reglas, el sistema se vuelve pesado y complejo.
- Posibles conflictos entre reglas (necesidad de mecanismos de resolución).
- Difícil capturar conocimiento experto completo en reglas simples.
- No maneja bien la incertidumbre, a menos que se combine con lógica difusa o probabilística.

4. Inferencia con lógica difusa (Fuzzy logic)

La lógica difusa (fuzzy logic) es una extensión de la lógica clásica que permite trabajar con grados de verdad en lugar de valores absolutos (solo verdadero/falso).

Mientras que en la lógica clásica una proposición puede ser 0 o 1, en lógica difusa puede tener cualquier valor entre 0 y 1, representando el grado de pertenencia a un conjunto.

Idea central

- En el mundo real, muchos conceptos son difusos:
 - Ejemplo: "alto", "caliente", "rápido", "leve dolor".
- La lógica clásica no puede expresarlos bien (o eres alto o no lo eres).
- La lógica difusa permite representarlos en forma gradual.

- Ejemplo: alguien de 1.80 m puede ser "alto con grado 0.8", y alguien de 1.70 m "alto con grado 0.5".

Componentes de un sistema de inferencia difusa

Un **sistema difuso** tiene 4 etapas principales:

1. Fuzzificación

- Convierte entradas numéricas (hechos concretos) en valores difusos.
- Usa **funciones de pertenencia** (triangulares, trapezoidales, gaussianas, etc.).
- Ejemplo: temperatura 37.8°C → "fiebre leve (0.6)", "fiebre alta (0.2)".

2. Base de reglas difusas

- Contiene reglas de producción en forma difusa, por ejemplo:
 - *Si temperatura es alta Ytos es leve → ENTONCES probabilidad de gripe es media.*
 - *Si temperatura es muy alta Y sarpullido es fuerte → ENTONCES probabilidad de sarampión es alta.*

3. Motor de inferencia difusa

- Combina hechos fuzzificados con las reglas para obtener conclusiones difusas.
- Aplica operadores lógicos difusos:
 - AND (mínimo), OR (máximo), NOT (complemento), entre otros.

4. Defuzzificación

- Convierte la salida difusa en un valor numérico concreto.
- Métodos comunes:
 - Centroide (más usado): calcula el centro de gravedad de la función.
 - Máxima pertenencia: selecciona el valor con mayor grado.

Ventajas

- Maneja incertidumbre e imprecisión, muy común en problemas reales.
- Se acerca al razonamiento humano natural (trabajamos con “más o menos”, no con absolutos).
- No necesita modelos matemáticos exactos (útil cuando los sistemas son muy complejos).
- Flexible y fácil de combinar con sensores y controladores.

Desventajas

- El diseño de las funciones de pertenencia y las reglas depende mucho del experto humano.

- Puede volverse complejo con muchas variables.
- No garantiza siempre la optimalidad matemática, es más heurístico.

5. Inferencia probabilística

La inferencia probabilística es un método de razonamiento que utiliza la teoría de la probabilidad para manejar incertidumbre y estimar la probabilidad de hipótesis o conclusiones dado un conjunto de hechos, síntomas o evidencias.

La inferencia probabilística es uno de los métodos más importantes de razonamiento en sistemas expertos modernos, especialmente cuando hay incertidumbre y falta de información completa. Es la base de muchos sistemas de diagnóstico, predicción y toma de decisiones.

Comparación: Inferencia probabilística vs. Inferencia difusa

Naturaleza de la incertidumbre

- **Probabilidad:**

- Maneja incertidumbre por falta de información.
- Ejemplo: “La probabilidad de que un paciente con fiebre tenga gripe es 70%”.
- Se enfoca en qué tan probable es que ocurra un evento.

- **Lógica difusa:**

- Maneja incertidumbre por vaguedad o imprecisión de conceptos.
- Ejemplo: “El paciente tiene fiebre *alta con grado 0.8*”.
- Se enfoca en qué tanto pertenece un valor a una categoría difusa.

Fundamento teórico

Se basa en:

1. **Probabilidad condicional:**

- $P(A | B)$ = probabilidad de A dado que B ocurre.
- Ejemplo: $P(\text{Gripe} | \text{Fiebre}) = 0.7$

2. **Teorema de Bayes:**

- Permite actualizar creencias con nueva evidencia:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

- Ejemplo: si un paciente tiene tos (E), actualizamos la probabilidad de que tenga gripe (H).

Cómo funciona paso a paso

- Definición de hipótesis posibles:** (ejemplo: gripe, COVID, resfriado).
- Asignación de probabilidades iniciales (priors):** se definen según estadísticas o experiencia.
 - Ejemplo: $P(\text{Gripe})=0.3$, $P(\text{COVID})=0.1$, $P(\text{Resfriado})=0.6$.
- Recolección de evidencia:** se obtienen síntomas o hechos (ejemplo: fiebre, tos).
- Aplicación de reglas probabilísticas:**
 - Se calcula $P(H|E)$ usando probabilidades condicionales.
 - Se actualizan las probabilidades de cada hipótesis.
- Conclusión:** se selecciona la hipótesis con mayor probabilidad o se muestran todas con sus valores.

Ventajas

- Permite razonar con incertidumbre de manera formal.
- Se basa en fundamentos matemáticos sólidos.
- Flexible: se puede actualizar con nueva evidencia.
- Muy útil en diagnóstico médico, predicciones, detección de fraudes, análisis de riesgos.

Desventajas

- Requiere conocer o estimar probabilidades previas y condicionales, lo cual puede ser difícil.
- Los cálculos se vuelven muy complejos cuando hay muchas variables.
- Supone que los eventos tienen una estructura probabilística bien definida.

Redes probabilísticas (o redes bayesianas):

Una red probabilística es un modelo gráfico que representa:

1. Variables aleatorias (hechos, síntomas, causas, eventos).
2. Relaciones de dependencia probabilística entre esas variables.
3. Una distribución conjunta de probabilidades que describe todo el sistema.

Se suelen representar como **grafos dirigidos acíclicos (DAG)**:

- **Nodos** = variables aleatorias.
- **Arcos (flechas)** = dependencia causal/probabilística.
- Cada nodo tiene una tabla de probabilidad condicional (CPT) que cuantifica la relación.

- Son un modelo gráfico que representa variables aleatorias y sus dependencias probabilísticas mediante un grafo dirigido.
- Permiten organizar y estructurar la información probabilística de forma visual y computacional.
- Facilitan el cálculo de inferencias probabilísticas complejas cuando hay muchas variables.

Relación con la inferencia probabilística

- La inferencia probabilística es el razonamiento.
- La red probabilística es la herramienta o modelo que organiza la información para hacer esa inferencia.

Fundamento matemático

Una red probabilística modela la distribución de probabilidad conjunta de un conjunto de variables X_1, X_2, \dots, X_n , usando la regla de factorización:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i))$$

Donde:

- $Pa(X_i)$ son los padres (variables que influyen) del nodo X_i .

Esto reduce un problema complejo en múltiples probabilidades pequeñas y manejables.

Componentes

1. **Nodos:** representan variables aleatorias (pueden ser discretas o continuas).
 - Ejemplo: “Gripe”, “Fiebre”, “Tos”.
2. **Arcos dirigidos:** indican relaciones de dependencia.
 - Ejemplo: $Gripe \rightarrow Fiebre$ (la gripe causa fiebre).
3. **Tablas de probabilidad condicional (CPT):** cada nodo tiene una tabla que indica la probabilidad de sus estados dados sus padres.
 - Ejemplo:
 - $P(Fiebre | Gripe) = 0.8$
 - $P(Fiebre | No Gripe) = 0.2$

Tipos de inferencia en redes probabilísticas

1. Diagnóstico (de efectos a causas):

- Ejemplo: ver fiebre y tos → calcular probabilidad de Gripe.

2. Predicción (de causas a efectos):

- Ejemplo: saber que hay Gripe → calcular probabilidad de tener fiebre.

3. Explicación intercausal:

- Ejemplo: si sé que COVID puede causar fiebre, la presencia de Gripe afecta menos mi creencia en COVID.

4. Actualización de creencias:

- Al añadir nueva evidencia, las probabilidades se actualizan automáticamente en toda la red.

Ventajas

- Manejan incertidumbre de manera formal.
- Reducen cálculos complejos mediante factorización.
- Permiten razonamiento causal (causas → efectos).
- Fácil de actualizar con nueva evidencia.
- Ampliamente aplicables en medicina, ingeniería, finanzas, IA y robótica.

Desventajas

- Requieren definir muchas probabilidades condicionales (puede ser costoso).
- Construcción de la red depende de expertos y datos estadísticos.
- Con muchas variables, la inferencia puede volverse computacionalmente costosa (NP-difícil en el peor caso).

6. Inferencia por casos (Case-based reasoning)

La inferencia por casos es un método de razonamiento que se basa en resolver problemas nuevos usando soluciones de problemas similares del pasado.

En lugar de aplicar reglas o probabilidades, el sistema:

1. **Busca casos previos almacenados** en una base de casos.
2. **Compara** el nuevo problema con esos casos.
3. **Adapta la solución antigua** para aplicarla al problema actual.

Componentes principales

1. **Base de casos:** almacena experiencias pasadas (problemas + soluciones).

- Ejemplo: pacientes con síntomas + diagnóstico final.
2. **Mecanismo de recuperación:** busca en la base de casos los más parecidos al problema actual.
 3. **Mecanismo de adaptación:** modifica la solución recuperada para ajustarla a las particularidades del nuevo caso.
 4. **Mecanismo de aprendizaje:** guarda el nuevo caso y su solución en la base para usos futuros.

Ciclo de razonamiento por casos (4R)

Se suele resumir en el ciclo de las 4R:

1. **Retrieve (Recuperar):**
Buscar en la base de casos los más similares al problema actual.
2. **Reuse (Reutilizar):**
Usar la solución del caso recuperado como base para resolver el problema.
3. **Revise (Revisar):**
Ajustar/adaptar la solución para el nuevo caso y verificar si funciona.
4. **Retain (Retener):**
Guardar el nuevo caso con su solución en la base de casos, enriqueciendo el sistema.

Ventajas

- Muy intuitivo y natural (similar al razonamiento humano).
- Permite aprender automáticamente con cada nuevo caso.
- No requiere una base de reglas exhaustiva ni modelos matemáticos complejos.
- Se adapta bien a dominios donde la experiencia pasada es muy valiosa (medicina, derecho, servicio técnico).

Desventajas

- Requiere una base de casos amplia y bien organizada.
- Puede ser ineficiente si no hay mecanismos de indexación/búsqueda adecuados.
- Depende mucho de la calidad y representatividad de los casos almacenados.
- La **adaptación de soluciones** no siempre es trivial.

7. Inferencia semántica / ontológica

La inferencia semántica/ontológica es el proceso de razonar a partir de relaciones semánticas y jerárquicas definidas en una ontología (un modelo formal del conocimiento de un dominio).

Una ontología organiza el conocimiento en:

- **Clases (conceptos):** Ej. *Enfermedad, Animal, Vehículo.*
- **Instancias (ejemplos):** Ej. *COVID-19 es una Enfermedad.*
- **Propiedades (relaciones):** Ej. *Enfermedad causa Síntoma, Perro es un Animal.*
- **Restricciones lógicas:** Ej. *toda Enfermedad infecciosa es una Enfermedad.*

La inferencia se produce al aplicar razonadores ontológicos que descubren nuevo conocimiento implícito a partir de lo explícito.

Fundamento teórico

La base está en la **lógica descriptiva (Description Logics, DL)**, que permite:

1. **Clasificación:** organizar automáticamente conceptos en jerarquías.
2. **Inferencia semántica:** deducir hechos no explícitos.
3. **Consistencia:** verificar que no haya contradicciones en la base de conocimiento.

Ejemplo en ontología:

- *COVID-19 es una Enfermedad Infecciosa.*
 - *Toda Enfermedad Infecciosa es una Enfermedad.*
- Se puede inferir automáticamente que *COVID-19 es una Enfermedad.*

Cómo funciona

1. **Se define la ontología:** conceptos, relaciones y axiomas.
2. **Se introducen instancias:** hechos concretos del dominio.
3. **El motor de inferencia semántica (razonador):** aplica las reglas de la lógica descriptiva.
4. **Se obtienen conclusiones:** se descubren nuevas relaciones o se validan consistencias.

Ventajas

- Maneja significado y contexto, no solo datos o probabilidades.
- Permite razonamiento automático en dominios complejos.
- Garantiza consistencia lógica en la base de conocimiento.
- Muy útil para sistemas que necesitan interoperabilidad semántica (ej. en la Web Semántica).

Desventajas

- Requiere construir una ontología formal (trabajo intensivo de expertos en conocimiento).
- Puede ser computacionalmente costoso en grandes dominios.
- No maneja bien incertidumbre cuantitativa (a diferencia de la probabilidad o la lógica difusa).

8. Inferencia híbrida / neuro-simbólica (moderna)

La inferencia híbrida/neuro-simbólica es un enfoque muy actual e innovador en inteligencia artificial porque busca combinar lo mejor de dos mundos:

- La **IA simbólica** (basada en reglas, lógica y conocimiento explícito).
- La **IA conexiónista** (redes neuronales, aprendizaje automático).

La inferencia híbrida/neuro-simbólica es un método de razonamiento en sistemas expertos e IA que integra:

1. **Métodos simbólicos:** reglas de producción, lógica formal, ontologías.
 - Pros: interpretables, consistentes, fáciles de explicar.
 - Contras: rígidos, requieren mucho conocimiento experto, no aprenden solos.
2. **Métodos conexiónistas:** redes neuronales, aprendizaje profundo.
 - Pros: aprenden patrones complejos, toleran ruido, generalizan bien.
 - Contras: cajas negras, difícil de explicar.

Al fusionarlos, se obtiene un sistema capaz de aprender de datos como una red neuronal y, al mismo tiempo, razonar y explicar como un sistema experto simbólico.

Fundamento teórico

El razonamiento híbrido busca unir:

- **Inferencia simbólica (top-down):** aplica reglas lógicas u ontológicas.
- **Inferencia conexiónista (bottom-up):** reconoce patrones desde ejemplos y datos.

La combinación se puede lograr de varias formas:

1. **Neuro-simbólica integrada:**
 - Las redes neuronales se entrena para generar representaciones que luego se interpretan con reglas simbólicas.
2. **Neuro-simbólica secuencial:**
 - Una red neuronal produce hipótesis que se validan con un motor simbólico.
3. **Neuro-simbólica acoplada:**
 - El sistema simbólico guía o restringe el aprendizaje de la red neuronal.

Ventajas

- **Explicabilidad:** las reglas simbólicas permiten justificar decisiones (cosa que las redes solas no hacen).
- **Aprendizaje automático:** las redes neuronales permiten que el sistema se entrene con datos masivos.
- **Robustez:** combina razonamiento lógico con reconocimiento de patrones.
- **Flexibilidad:** se adapta a dominios con conocimiento explícito (leyes, medicina) y datos no estructurados (imágenes, textos).

Desventajas

- Complejidad de implementación (se necesitan expertos en ambos enfoques).
- Mayor costo computacional.
- Difícil mantener equilibrio entre “rigidez lógica” y “flexibilidad neuronal”.

Reglas de inferencia Modus Ponens y Modus Tollens

El Modus Ponens y el Modus Tollens son más bien reglas elementales de inferencia que forman parte de los métodos, en especial en el razonamiento deductivo.

Qué es Modus Ponens?

Es una regla de inferencia deductiva muy usada en sistemas expertos basados en reglas:

1. Si $P \rightarrow Q$ (si P entonces Q).
2. P es verdadero.
3. Entonces Q es verdadero.

👉 Ejemplo en un sistema experto médico:

- Regla: *Si el paciente tiene fiebre \rightarrow entonces el paciente está enfermo.*
- Hecho: *El paciente tiene fiebre.*
- Conclusión: *El paciente está enfermo.*

¿Qué es Modus Tollens?

También se usa en inferencia lógica:

1. Si $P \rightarrow Q$.
2. $\neg Q$ (Q es falso).
3. Entonces $\neg P$ (P es falso).

Ejemplo:

- Regla: *Si el coche tiene gasolina → entonces arranca.*
- Hecho: *El coche no arranca.*
- Conclusión: *El coche no tiene gasolina.*

Esquema jerárquico de los métodos de inferencia

Nivel 1: Reglas básicas de inferencia (lógica proposicional)

Son los bloques fundamentales del razonamiento:

- **Modus Ponens** → Si $P \rightarrow Q$, y P es cierto → entonces Q .
- **Modus Tollens** → Si $P \rightarrow Q$, y Q es falso → entonces P es falso.
- **Silogismo hipotético** → Si $P \rightarrow Q$, y $Q \rightarrow R$ → entonces $P \rightarrow R$.
- **Silogismo disyuntivo** → $P \vee Q$, y $\neg P$ → entonces Q .

Estas reglas son el “motor lógico” sobre el que se construyen los métodos más complejos.

Nivel 2: Estrategias de inferencia clásica (IA simbólica)

Usan reglas básicas para ejecutar razonamiento más amplio:

- **Encadenamiento hacia adelante (forward chaining)** → aplica Modus Ponens repetidamente desde los hechos iniciales.
- **Encadenamiento hacia atrás (backward chaining)** → aplica Modus Tollens y búsqueda de metas para verificar hipótesis.
- **Razonamiento deductivo, inductivo y abductivo.**

Nivel 3: Métodos de inferencia con incertidumbre

Extienden las reglas clásicas para manejar información incompleta o ambigua:

- **Lógica difusa (fuzzy logic)** → permite grados de verdad.
- **Inferencia probabilística / bayesiana** → usa probabilidades condicionales en lugar de valores absolutos.
- **Redes neuronales probabilísticas** → mezclan patrones estadísticos con razonamiento lógico.

Nivel 4: Métodos basados en experiencia y semántica

Aplican inferencia sobre estructuras más ricas de conocimiento:

- **Case-Based Reasoning (CBR)** → usa experiencias pasadas.
- **Inferencia semántica/ontológica** → usa jerarquías y relaciones de ontologías (RDF, OWL).

Nivel 5: Métodos híbridos y modernos (neuro-simbólicos)

Combinan lógica + aprendizaje automático:

- **Motores híbridos** → aplican reglas SI-ENTONCES con soporte de modelos estadísticos.
- **Inferencia neuro-simbólica** → redes neuronales generan hipótesis, y las reglas lógicas las validan/explican.
- **IA explicable (XAI)** → mezcla modelos de ML con razonamiento lógico para dar explicaciones claras.