



Trabajo Práctico N°2: Razonamiento

Inteligencia Artificial II

Francisco Castel
Melanie Martinez
Brian Nuñez
Juan Ignacio Quiroga
Nicolas Sorrentino
Ingeniería en Mecatrónica

20 de abril de 2025

Resumen

Este trabajo práctico presenta las respuestas a las preguntas teóricas sobre razonamiento en Inteligencia Artificial, incluyendo algoritmos basados en inferencia lógica, bases de conocimiento, lógica proposicional, lógica de primer orden y planificación.

1. ¿Qué motivación tenemos para utilizar algoritmos basados en la inferencia lógica en lugar de utilizar algoritmos de búsqueda para cualquier tipo de problema?

En el ámbito de la inteligencia artificial, los algoritmos de búsqueda y los métodos basados en inferencia lógica son herramientas fundamentales, pero su aplicabilidad varía según la naturaleza del problema a resolver.

Algoritmos de búsqueda como A*, BFS o DFS son eficaces cuando se puede definir un espacio de estados claro y finito, con reglas de transición bien establecidas. Son ideales para problemas donde se busca una solución óptima a través de la exploración de diferentes caminos, como en juegos o planificación de rutas.

Sin embargo, cuando enfrentamos problemas que requieren razonamiento simbólico, manejo de conocimiento complejo o deducción a partir de hechos y reglas, los algoritmos de búsqueda pueden resultar ineficientes o inadecuados. Aquí es donde los algoritmos basados en inferencia lógica ofrecen ventajas significativas:

1. **Representación declarativa del conocimiento:** Permiten modelar el conocimiento del dominio mediante hechos y reglas lógicas, facilitando la comprensión y mantenimiento del sistema.
2. **Inferencia a partir de información incompleta:** Pueden deducir nueva información incluso cuando no se dispone de todos los datos, lo cual es esencial en entornos dinámicos o parcialmente observables.
3. **Reutilización del conocimiento:** La base de conocimientos puede ser utilizada para múltiples propósitos, como diagnóstico, planificación o toma de decisiones, sin necesidad de redefinir el problema.
4. **Transparencia y explicabilidad:** Las conclusiones derivadas pueden ser trazadas hasta los hechos y reglas que las generaron, lo que es crucial en aplicaciones donde se requiere justificar las decisiones tomadas.

Ejemplo práctico:

Consideremos un sistema experto en medicina. Utilizando inferencia lógica, podemos representar síntomas y enfermedades mediante reglas como:

- Si el paciente tiene fiebre y tos, entonces podría tener gripe.
- Si el paciente tiene erupciones cutáneas y fiebre, entonces podría tener sarampión.

A partir de los síntomas observados, el sistema puede inferir posibles diagnósticos, incluso si no se presentan todos los síntomas típicos de una enfermedad. Además, puede explicar al médico cómo llegó a esa conclusión, lo que no sería posible de manera directa con un algoritmo de búsqueda.

En resumen, la elección entre algoritmos de búsqueda e inferencia lógica depende del tipo de problema. Para tareas que requieren razonamiento complejo, manejo de conocimiento simbólico y explicabilidad, la inferencia lógica es la herramienta más adecuada.

2. ¿Qué diferencia hay entre una base de conocimientos y el ground truth?

En un sistema inteligente, especialmente en agentes basados en conocimiento, es importante distinguir entre la base de conocimientos (KB) y el ground truth, ya que ambos representan tipos distintos de información y cumplen funciones complementarias.

- **Base de Conocimiento (KB):** Es el componente central que almacena conocimiento del dominio en forma de hechos, reglas, relaciones y axiomas. Está compuesta por:
 - **Background knowledge:** conocimiento general del dominio, que suele ser estable en el tiempo. Ejemplo: “una válvula puede estar abierta o cerrada”.
 - **Ground facts:** hechos particulares de una instancia concreta del problema. Ejemplo: “la válvula A está cerrada”.
 - Esta información permite al agente realizar inferencias, planificar y tomar decisiones. No es necesariamente inmutable, ya que puede actualizarse con nuevos hechos.
- **Ground Truth:** Se refiere a los datos reales del mundo, recolectados por sensores o fuentes externas confiables. Representa el estado actual del entorno, y sirve como referencia para validar, ajustar o alimentar modelos de razonamiento.
 - Es mutable, ya que puede cambiar constantemente en función de lo que percibe el agente.
 - Puede ser incorporado a la KB como hechos (ground facts), para que el sistema pueda operar sobre ellos con sus reglas de inferencia.

En síntesis, la base de conocimientos permite razonar y planificar usando reglas y relaciones del dominio, mientras que el ground truth representa lo que realmente está pasando en el entorno en un momento dado. Ambos son necesarios para que el agente actúe de forma coherente y adaptativa.

3. ¿Qué diferencia hay entre un valor inferido y un valor percibido?

Valor Percibido:

El valor percibido es aquel que el agente obtiene directamente desde el entorno mediante sensores o cualquier otro dispositivo de adquisición de datos. Se trata de información observada o medida en tiempo real, la cual es concreta y refleja directamente el estado actual del mundo en el momento en que es tomada la medición.

- **Características principales:**
 - Proviene directamente de sensores.
 - Es inmediata y refleja la realidad observada en tiempo real.
 - Puede ser ruidosa o imprecisa debido a limitaciones en los sensores.

■ Ejemplos prácticos:

- La lectura instantánea de la temperatura medida por un termómetro.
- El estado actual de una válvula (abierta o cerrada), detectado por un sensor.

Valor Inferido:

El valor inferido no es obtenido directamente desde los sensores, sino que se calcula o deduce a partir de una base de conocimientos utilizando reglas de inferencia y lógica. Surge del razonamiento lógico realizado por el agente con base en hechos, axiomas y relaciones ya definidas.

■ Características principales:

- Se obtiene mediante procesos de razonamiento lógico.
- Se deriva a partir de información existente (background knowledge, reglas, ground facts).
- Permite completar o enriquecer información no directamente observable.

■ Ejemplos prácticos:

- Si conocemos que “una tubería explota si la presión supera los 100 psi”, y percibimos una presión de 110 psi, inferimos que la tubería explotará.
- Dado que “la presión en la tubería A es igual a la presión en la tubería B más la mitad de la presión en la tubería C”, y conocemos valores percibidos de presión en B y C, podemos inferir la presión en A aunque no la midamos directamente.

Diferencias Clave:

Aspecto	Valor Percibido	Valor Inferido
Fuente	Sensores, medición directa	Razonamiento lógico a partir de reglas
Precisión	Puede tener ruido o imprecisión	Depende de la validez del razonamiento
Rapidez de adquisición	Inmediata, en tiempo real	Requiere un proceso lógico previo
Ejemplo	Temperatura medida con un termómetro	Temperatura calculada mediante reglas

Relación entre ambos conceptos:

Generalmente, el agente inteligente combina ambos tipos de información:

- Primero utiliza los valores percibidos (entrada sensorial).
- Luego aplica reglas lógicas para derivar valores inferidos y obtener así un entendimiento más completo del estado del sistema o del entorno.

Esta combinación permite que el agente actúe de manera más robusta y eficiente, especialmente en situaciones con información parcial o incertidumbre.

4. ¿Qué valores puede tomar una oración atómica de acuerdo a la lógica proposicional? ¿Cómo se modela la frase “Los valores de A comprendidos entre 0 y 9” de acuerdo a esta sintaxis?

1. Valores que puede tomar una oración atómica en lógica proposicional:

Una oración atómica en lógica proposicional solo puede tomar dos valores posibles:

- Verdadero (V)
- Falso (F)

Estos valores son estrictamente dicotómicos; es decir, cada proposición solo puede ser verdadera o falsa, sin admitir estados intermedios.

2. Modelado de la frase “Los valores de A comprendidos entre 0 y 9”:

En lógica proposicional, para expresar que una variable A puede tomar valores desde 0 hasta 9, se suele utilizar una disyunción (operador lógico OR) para indicar que al menos uno de los valores posibles debe cumplirse. Si quisiéramos expresar que A está específicamente comprendido en ese rango (al menos uno de esos valores es verdadero), lo haríamos así:

$$A_0 \vee A_1 \vee A_2 \vee A_3 \vee A_4 \vee A_5 \vee A_6 \vee A_7 \vee A_8 \vee A_9$$

Esto significa que al menos uno de los valores (desde A_0 hasta A_9) debe ser verdadero para que la expresión completa sea verdadera.

5. ¿Qué algoritmos se utilizan para resolver problemas modelados con lógica proposicional?

En lógica proposicional, se emplean diversos enfoques y algoritmos para determinar la satisfacibilidad de fórmulas, inferir consecuencias lógicas o automatizar demostraciones. A continuación, se presentan los principales:

1. Algoritmos de Inferencia Lógica

Estos algoritmos permiten deducir conclusiones a partir de un conjunto de proposiciones conocidas (la base de conocimientos).

- **Encadenamiento hacia adelante:** parte de los hechos conocidos y aplica reglas para deducir nuevas verdades.
- **Encadenamiento hacia atrás:** parte de una meta o consulta y busca si puede demostrarse usando los hechos y reglas disponibles.

Estos métodos pueden representarse como búsquedas en grafos, donde los nodos son proposiciones y los arcos representan implicaciones lógicas.

2. Método de Resolución

Es un procedimiento formal para comprobar si un conjunto de proposiciones es satisfacible (es decir, si puede existir una asignación de valores de verdad que haga que todas sean verdaderas simultáneamente).

■ Procedimiento:

- Convertir todas las fórmulas a la forma normal conjuntiva (FNC).
- Aplicar la regla de resolución para generar nuevas cláusulas.
- Si se deriva la cláusula vacía, el conjunto original es insatisfacible.

■ Ventajas:

- Es un método completo y sistemático para la lógica proposicional.

■ Limitaciones:

- Puede ser ineficiente para conjuntos grandes debido al crecimiento exponencial de cláusulas.

3. Solucionadores de SAT

Para problemas más grandes o reales, se utilizan algoritmos más sofisticados que automatizan la verificación de satisfacibilidad, como:

- **DPLL (Davis–Putnam–Logemann–Loveland):** algoritmo de búsqueda con backtracking, eliminación de literales puros y propagación de unidades.
- **SAT Solvers modernos** (como MiniSAT, Glucose, zChaff): utilizan técnicas como aprendizaje de cláusulas, análisis de conflictos y backtracking no cronológico para resolver problemas de satisfacibilidad de gran escala.

6. ¿Qué valores puede tomar una oración atómica de acuerdo a la lógica de primer orden? ¿Cómo se modela la frase “Los valores de A comprendidos entre 0 y 9” de acuerdo a esta sintaxis?

1. Valores de una oración atómica en Lógica de Primer Orden:

Al igual que en la lógica proposicional, una oración atómica en LPO solo puede tomar dos valores posibles:

- Verdadero (V)
- Falso (F)

Lo que cambia no es el tipo de valor, sino la estructura y expresividad de las oraciones. En LPO se pueden representar objetos, relaciones entre ellos y propiedades, mediante predicados, funciones y constantes.

2. Modelado de “Los valores de A comprendidos entre 0 y 9” en LPO:

En lugar de listar individualmente cada proposición como en la lógica proposicional ($A_0 \vee A_1 \vee \dots \vee A_9$), en lógica de primer orden usamos predicados con variables que permiten representar esta relación de manera más compacta y general.

$$\text{valor_permitido}(A) \leftarrow A \geq 0 \wedge A \leq 9$$

Ejemplo de modelado con predicados:

Aspecto	Lógica Proposicional	Lógica de Primer Orden
Representación	Proposiciones aisladas (e.g., A_0, A_1, \dots)	Predicados con variables (e.g., $\text{valor_permitido}(A)$)
Nivel de abstracción	Bajo	Alto
Cantidad de oraciones	Exponencial con el dominio	Compacta
Modelado de relaciones	Limitado o forzado	Natural y expresivo

La lógica de primer orden permite representar el conocimiento de manera más general y reutilizable, evitando la explosión combinatoria que se genera en lógica proposicional. Esto la hace ideal para modelar dominios complejos con múltiples objetos y relaciones.

7. ¿Qué algoritmos se utilizan para resolver problemas modelados con lógica de primer orden?

La lógica de primer orden (LPO) es mucho más expresiva que la proposicional, ya que permite representar objetos, relaciones y cuantificadores. Sin embargo, esta mayor expresividad conlleva también un aumento en la complejidad computacional.

Para resolver problemas modelados con LPO, se utilizan principalmente algoritmos de inferencia lógica, que pueden clasificarse en dos grandes enfoques: algorítmicos generales y específicos por lenguaje o técnica.

1. Algoritmos de Inferencia Basados en Reglas

Son técnicas que aplican reglas lógicas para derivar conclusiones a partir de una base de conocimientos. Los más comunes son:

■ Encadenamiento hacia adelante (forward chaining):

- Comienza con los hechos conocidos (ground facts) y aplica reglas para inferir nuevos hechos.
- Muy usado en sistemas expertos.

■ Encadenamiento hacia atrás (backward chaining):

- Parte de una meta y busca demostrarla con hechos existentes y reglas disponibles.
- Es la base del funcionamiento de lenguajes como Prolog.

2. Algoritmos de Unificación

- La unificación es un proceso fundamental en LPO y en lenguajes como Prolog.
- Permite encontrar una sustitución de variables que haga coincidir dos expresiones lógicas.
- Es esencial para aplicar reglas en inferencia lógica, ya que permite comparar predicados con variables.

3. Algoritmo de Resolución para LPO

- Es una generalización del método de resolución proposicional, adaptado para manejar cuantificadores y predicados.
- Funciona sobre la forma normal de Skolem (cláusulas de Horn) y aplica resolución binaria para deducir consecuencias.
- Es completo: si algo se puede demostrar lógicamente a partir de la base de conocimiento, este método puede hacerlo.
- Sin embargo, no es decidible en general: puede no terminar si el problema es irresoluble o infinitamente profundo.

4. Motores de Inferencia de Prolog

- Prolog utiliza resolución basada en encadenamiento hacia atrás y unificación para resolver consultas.
- Su motor de inferencia explora recursivamente la base de conocimientos tratando de satisfacer una consulta, aplicando reglas y unificando términos.
- El algoritmo implementa una búsqueda en profundidad con backtracking, lo que permite navegar por las posibles soluciones.

5. Técnicas adicionales:

- **Demostración automática de teoremas:** se aplican estrategias como resolución con estrategias de control, saturación, árboles semánticos, etc.
- **Métodos heurísticos:** en problemas muy grandes o del mundo real, pueden usarse heurísticas para acotar el espacio de búsqueda o priorizar reglas.

Enfoque / Algoritmo	Aplicación principal
Encadenamiento hacia adelante	Inferencia automática desde hechos conocidos
Encadenamiento hacia atrás	Verificación de metas / consultas (base de Prolog)
Unificación	Comparación y emparejamiento de términos con variables
Resolución para LPO	Demostración lógica a partir de reglas y predicados
Motor Prolog	Ejecución práctica de inferencia basada en LPO

8. Describa cómo los planificadores hacen uso de algoritmos de búsqueda globales para encontrar una secuencia óptima de pasos a seguir para resolver un problema

Los planificadores trabajan modelando un problema como un sistema que tiene:

- Estado inicial
- Conjunto de operadores (acciones)
- Meta u objetivo (estado deseado)

Para encontrar una solución, el planificador genera una secuencia de operadores (plan) que, aplicados al estado inicial, conduzcan al estado objetivo.

Este proceso se realiza mediante algoritmos de búsqueda global, que exploran el espacio de estados definido por todas las posibles aplicaciones de operadores.

¿Qué es el espacio de estados?

Es la representación de todos los estados alcanzables a partir del estado inicial aplicando distintas combinaciones de acciones permitidas. Cada nodo en este espacio representa un estado, y cada arco representa una acción.

¿Cómo se integran los algoritmos de búsqueda global?

Los algoritmos de búsqueda global se usan para recorrer el espacio de estados. En la guía de resolución se ejemplifica con el uso de operadores instanciados (como cargar(FERTILIZANTE, LA02, AEP)) que generan nuevos estados a partir de condiciones previas cumplidas.

Etapas del proceso:

1. Instanciación de operadores:

- Se aplican reglas de planificación tipo STRIPS o PDDL que definen:
 - Precondiciones para aplicar una acción.
 - Efectos de esa acción sobre el estado actual.
- Por ejemplo:

```
(:action cargar
:parameters (?c ?a ?ap)
:precondition (and (en ?c ?ap) (en ?a ?ap) ...)
:effect (and (en ?c ?a) (not (en ?c ?ap)))
)
```

2. Construcción del grafo de búsqueda:

- Cada vez que se aplica una acción válida se genera un nuevo estado (nuevo nodo del grafo).

3. Selección de caminos mediante heurísticas:

- Se puede usar una búsqueda ciega (como DFS o BFS) o una búsqueda informada, como A^* , para encontrar la secuencia óptima de acciones que minimicen el costo total (número de pasos, tiempo, recursos, etc.).

Los planificadores usan algoritmos de búsqueda global para explorar el espacio de estados generado por la aplicación de operadores, buscando una secuencia válida y óptima de acciones que cumplan los objetivos. Esto se logra modelando adecuadamente precondiciones y efectos, instanciando operadores, y utilizando un algoritmo de búsqueda (como A^*) que seleccione la mejor trayectoria posible.