



Tarea 3

Robin Emmanuel Carlos Gonzalez 21110427

Motores de inferencia de la actualidad

1) Motores lógicos / de reglas (expert systems, KBS, BRMS)

Aplican reglas tipo IF-THEN (Rete, forward/backward chaining) para deducir hechos y disparar acciones. Útiles cuando tienes conocimiento explícito y auditable.

- Drools (Java): motor de reglas empresarial muy usado (Rete, reglas + hechos).
- CLIPS (NASA) y Jess (Java): shells clásicos de sistemas expertos basados en reglas.
- Prolog / SWI-Prolog: lógica de predicados con resolución; base para sistemas con backtracking.
- Apache Jena (RDF/OWL): framework semántico con razonadores integrados para RDFS/OWL y reglas.

2) Razonadores de ontologías (Description Logics / OWL)

Verifican consistencia y clasifican ontologías (OWL 2), muy usados con grafos de conocimiento.

- HermiT, Pellet, FaCT++/JFact.
- RDFox, Stardog: razonamiento Datalog de alto rendimiento sobre RDF/OWL + reglas.

3) Motores de inferencia probabilística (PPL / gráficos bayesianos)

Cuando hay incertidumbre y necesitas estimar distribuciones/posteriores.

- PyMC, Stan, Pyro (sobre PyTorch)
- TensorFlow Probability / Edward2 integra modelos probabilísticos con TensorFlow.
- ProbLog: programación lógica probabilística.

4) Solvers lógicos y de restricciones (SAT/SMT/CP)

No “razonan” con reglas, pero infieren satisfacibilidad/Modelos; clave en verificación formal, planificación y síntesis.

- Z3 (Microsoft Research), cvc5 (SMT), MiniSat (SAT).

5) Motores de inferencia para redes neuronales (runtime/compiladores)

Optimizan y ejecutan modelos entrenados (latencia/throughput), desde móvil hasta servidor.

- ONNX Runtime, NVIDIA TensorRT, Intel OpenVINO, TensorFlow Lite, Apache TVM.

Métodos de inferencia

1) Lógicos / simbólicos (deterministas)

Modus Ponens / Modus Tollens:

- Encadenamiento hacia adelante / hacia atrás: dispara reglas con hechos (forward) o parte de una meta (backward).
- Resolución en CNF + unificación (Prolog/SLD): deducción por refutación; muy usado en lógica de primer orden.
- Tableaux semánticos / cálculo de secuentes: prueba de teoremas y chequeo de validez.
- Razonamiento en ontologías (DL/OWL, Datalog): clasificación, consistencia y materialización en grafos de conocimiento.

2) Probabilísticos (con incertidumbre)

- Bayes exacto: eliminación de variables, creencia/propagación de mensajes (sum-product), MAP/MPE; HMM/Viterbi, filtros de Kalman.
- MCMC (Gibbs, Metropolis–Hastings): muestreo cuando exacto es intratable.
- Inferencia variacional (mean-field, BBVI): aproxima la posterior optimizando el ELBO; rápida en grandes modelos.
- Redes bayesianas/Markov, PGM: estructuran dependencias para inferir eficientemente.

3) Difusa / borrosa

- Mamdani y Sugeno–Takagi: reglas IF–THEN con grados de verdad; defuzzificación (centroide, bisector). Útil con variables lingüísticas y sensores ruidosos.

4) Restricciones y satisfacibilidad

- CSP: backtracking + consistencias (AC-3), poda; agendas/horarios.
- SAT/SMT (DPLL/CDCL): comprobar si hay asignaciones que satisfacen fórmulas (verificación formal).
- Optimización (LP/ILP/CP-SAT): inferir la mejor solución sujeta a restricciones.
- CLP (Programación Lógica con Restricciones): mezcla lógica + dominios numéricos.

5) No monótona y por defecto

- Default reasoning, circunscripción, autoepistémica, ASP (Answer Set Programming): permiten retractar conclusiones ante nueva evidencia (conclusiones “derrotables”).

6) Abducción, analogía y casos

- Abducción: mejor explicación para una observación (diagnóstico de fallas).
- Razonamiento analógico y CBR (Case-Based Reasoning): adaptar soluciones de casos similares.

7) Causal

- Do-calculus (Pearl), ajuste backdoor/IV, gráficos causales: inferir efectos de intervenciones (qué pasa si “forzo” X).

8) “Inferencia” en modelos neuronales

- Pase hacia adelante (predicción determinista) y, si hay incertidumbre, redes bayesianas / MC-Dropout para estimar confianza.

Modus Ponens

Qué es:

Un razonamiento directo. Tienes una regla del tipo “si pasa A, entonces pasa B”. Cuando confirmas que A sí ocurrió, concluyes que B también ocurre.

Cómo se entiende:

Activas la condición y, por lo tanto, activas la consecuencia.

Ejemplos:

- Cotidiano: “Si el semáforo está en verde, avanzo”. El semáforo está en verde → avanzo.
- Ingeniería: “Si la temperatura supera 80 °C, el motor se apaga”. La temperatura superó 80 °C → el motor se apaga.
- Diagnóstico simple: “Si el usuario está autenticado, puede entrar al panel”. El usuario está autenticado → entra al panel.

Cuándo usarlo:

Cuando sabes que se cumple la condición de una regla confiable y necesitas disparar su efecto.

Error típico a evitar:

Creer que porque ves el efecto, necesariamente ocurrió la causa (eso es “afirmar el consecuente”, y es inválido).

Modus Tollens

Qué es:

Un razonamiento por descarte. Partes de una regla “si pasa A, entonces pasa B”. Si observas que B no ocurrió, concluyes que A no ocurrió.

Cómo se entiende:

Si A siempre trae B, entonces encontrar un caso sin B significa que no hubo A.

Ejemplos:

- Cotidiano: “Si hay internet, el ping responde”. El ping no responde → no hay internet.
- Ingeniería: “Si el sensor está conectado, el sistema muestra lectura”. No hay lectura → el sensor no está conectado.
- Calidad/Producción: “Si la pieza pasó calibración, cumple tolerancias”. No cumple tolerancias → no pasó calibración.

Cuándo usarlo:

Cuando no ves el efecto que una causa siempre produciría y quieres descartar esa causa.

Cuidado importante:

Funciona cuando la regla es estricta y sin excepciones. En la vida real, si puede haber otras causas para que falte el efecto, debes verificarlas (de lo contrario, tu conclusión puede ser apresurada).