

TEMA 3.

Representación del Conocimiento

Francisco José Ribadas Pena

INTELIGENCIA ARTIFICIAL
5 Informática
ribadas@uvigo.es

6 de noviembre de 2009

3.1 El Problema de la Represent. del Conocimiento

3.1.1 Aspectos Generales

- Punto clave en la I.A. simbólica:
 - Uso de conocimiento y de mecanismos para representarlo y manejarlo computacionalmente
 - Diferencia con informática convencional:
 - uso extensivo de conocimiento del dominio
- Hasta ahora (búsqueda espacios estados): representaciones "*ad-hoc*", específicas para cada problema concreto
 - conocimiento del dominio {
 - represent. de estados
 - represent. de operadores
 - función heurística

(a) Jerarquía del Conocimiento

- Datos: elementos de interés en un dominio (*datos en "bruto"*)
 - Ejemplo: letras, dígitos,
- Información: datos relevantes procesados
 - Ejemplo: nombre, fechas, temperaturas,...

- Conocimiento: información especializada de alto nivel (información interrelacionada)
 - Resultado de la experiencia de expertos
 - Organización e interrelaciones entre "piezas" de información
 - Ejemplo: ¿cuándo se considera alta una temperatura corporal?, ¿qué significa que además haya tos?, precios caros o baratos, ...
- Metaconocimiento: conocimiento a cerca del conocimiento
 - Indica cómo utilizar el conocimiento
 - Ejemplo: Conocimiento que permita decidir desechar parte del conocimiento que no sea relevante en determinadas situaciones (ej.: selección de heurísticas)

(b) Uso del conocimiento en I.A. simbólica

- *Sistemas Basados en el Conocimiento*: Sistemas de I.A. que hacen un uso extensivo del conocimiento específico de un dominio
 - **base de conocimiento (BC)**: conjunto de representaciones simbólicas de hechos y relaciones relevantes en un dominio
 - "acota" el mundo sobre el cual se razona
- Pasos Generales:
 1. Identificar conocimiento necesario para resolver un problema
 2. Seleccionar un lenguaje con el que representar ese conocim.
 3. Traducir conocimiento a ese lenguaje de representación [construir B.C.]
 4. Usar las consecuencias derivadas de ese conocimiento para resolver el problema [razonamiento]
 - estrategias de búsqueda en inferencia

3.1.2 Objetivos de los Métodos de Rep. del Conoc.

Representación del conocimiento

- DEF.: Proceso de transformación del conocimiento de un dominio a un lenguaje simbólico para ser procesado computacionalmente
 - Planteamiento simbólico de resolución de problemas (maneja símbolos)
 - En IA subsimbólica (RNAs, alg. genéticos) no existe relación directa entre el conocimiento y su representación
 - Cada elemento de la representación (símbolo) se refiere a un objeto/hecho/relación de interés perteneciente al dominio a representar
- Todo lenguaje de representación debe definir 2 aspectos
 - SINTAXIS: posibles formas de construir y combinar los elementos del lenguaje para representar los hechos del dominio real
 - SEMÁNTICA: determina la relación entre los elementos del lenguaje y su interpretación en el dominio
- Dos fases en la representación
 - FASE DE CODIFICACIÓN (representación): conversión de hechos reales a su representación interna
 - FASE DE DESCODIFICACIÓN (interpretación): los resultados de los procesos inferenciales realizados sobre la representación interna del conocimiento vuelven a convertirse en hechos del mundo real
- Esquema general

- Ejemplos: Representación de conocimiento empleando lógica de predicados y mediante redes semánticas

3.1.3 Propiedades de los Esquemas de Representación

- ADECUACIÓN DE LA REPRESENTACIÓN: Capacidad del esquema de representación para representar adecuadamente *todo* el conocimiento pertinente de un dominio
 - **Objetivo:** soportar cualquier tipo de conocimiento relevante
- ADECUACIÓN INFERENCIAL: Posibilidad de manipular las estructuras de representación de forma que se puedan derivar nuevas estructuras asociadas con nuevo conocimiento inferido a partir del antiguo
- EFICIENCIA INFERENCIAL: Posibilidad de mejora del proceso inferencial mediante la inclusión de heurísticas y guías que agilicen la inferencia
 - **Objetivo:** procesar conocimiento e inferir con un coste computacional aceptable
- EFICIENCIA ADQUISICIONAL: Capacidad del esquema para incorporar fácilmente nuevo conocimiento
 - de forma manual (usuario/desarrollador)
 - de forma automática (aprendizaje automático)
- Otras Características:
 - TRANSPARENCIA: Posibilidad de identificar fácilmente el conocimiento representado
 - poder interpretar directamente ("a ojo") que representa cada estructura
 - NATURALIDAD Y CLARIDAD: Posibilidad de representar el conocimiento en su forma original (conocimiento no es transformado)
 - Por ej.: almacenar directamente frases en lenguaje natural
 - MODULARIDAD: Capacidad del esquema para soportar la fragmentación del conocimiento sin perder eficiencia ni eficacia
 - GRANULARIDAD: Grado de detalle de una repres. del conocim.
 - depende de los requisitos del problema y del tipo de conocimiento a representar

3.1.1 Clasificación de los Esquemas de Representación

1. Esquemas declarativos: [→ punto clave: representar]

- Enfasis en representación de los hechos relevantes del dominio
- **Objetivo:** Organizar correctamente las entidades del dominio
 - ≈ bases de datos de alto nivel
- Es posible definir sistemas que sólo manejen conocimiento declarativo (sistemas de "consulta")
- Soportan tipos limitados de procedimientos de inferencia
- Conocimiento nuevo se obtiene por *ejemplificación* del antiguo
- Fácil incorporar conocimiento nuevo sin afectar al ya existente
- *Ejemplos:* marcos y redes semánticas (sistemas basados en la herencia), lógica formal

2. Esquemas procedimentales [→ punto clave: razonar]

- Enfasis en representar y soportar las relaciones inferenciales del dominio
- **Objetivo:** Aplicar métodos potentes de inferencia sobre conjuntos relativamente reducidos de hechos
- Permiten razonar y obtener inferencias con falta de conocimiento
- Adecuados para representar conocimiento heurístico (experiencia)
- *Ejemplos:* métodos búsqueda en espacio de estados, sistemas de reglas de producción

En la práctica: mezcla de ambas aproximaciones (reglas de producción, lógica formal)

Tipos de modelos de representación del conocimiento

1. Modelos formales (lógica clásica) [base teórica sólida]
 - Uso de la lógica formal como lenguaje de representación
 - Mecanismos de inferencia potentes, bien conocidos y formalizados
 - *Modus Ponens*
 - *Resolución*
 - *Ejemplos:*
 - Lógica clásica: lógica proposiciones y lógica predicados 1^{er} orden
 - Lógicas no clásicas: lógica modal, lógica temporal, lógica difusa
2. Modelos estructurados [modelos *ad hoc*]
 - Uso de técnicas de representación especializadas
 - Incorporan mecanismos de inferencia específicos, muy eficientes, pero limitados
 - *Ejemplos:*
 - Redes semánticas
 - Marcos y guiones
 - Sistemas basados en reglas de producción

3.2 Representaciones Formales. Lógica

- Método de representación del conocimiento bien conocido, fuertemente formalizado y muy usado
 - VENTAJAS: { base teórica muy sólida
mecanismos inferencia conocidos y potentes
- Toda lógica que sea usada como mecanismo de representación constará de 3 elementos:
 1. Una **sintaxis**, que explica cómo construir los elementos del lenguaje de representación (oraciones lógicas)
 2. Una **semántica**, que asocia los elementos del lenguaje con los elementos reales del dominio
→ asocia una *interpretación* (significado) a los símbolos
 3. Un conjunto de **reglas de inferencia**, que permiten inferir conocimiento nuevo a partir del ya existente [mediante transformaciones sintácticas]

RAZONAMIENTO EN LÓGICA FORMAL

- Uso de **razonamiento deductivo**
 - Una aseveración será cierta si se puede demostrar su veracidad a partir de observaciones que se sabe que son ciertas
→ todo lo que se infiere es verdadero
 - Pasar de conocimiento general a nuevo conocimiento específico

cuando hay gripe, tienes fiebre	Juan tiene gripe
Juan tiene gripe	Juan tiene fiebre

Juan tiene fiebre	cuando hay gripe tienes fiebre
(Razonam. deductivo)	[LOGICA]
(Razonam. inductivo)	[APRENDIZAJE]

Juan tiene fiebre	
cuando hay gripe, tienes fiebre	

Juan tiene gripe	
(Razonam. abductivo)	[RAZ. PROBABILISTICO]

→ Raz. inductivo y abductivo no aseguran que sus conclusiones sean siempre verdaderas

- IMPORTANTE: Mecanismo de *refutación* mediante *resolución*
 - Uso de la regla de inferencia de *resolución*
 - Procedimiento formal de prueba (demonstración) que aplicado sistemáticamente permite determinar si una sentencia es consecuencia lógica de otras sentencias [≈ B.C.]
 - Procedimiento general
 - Puede ser implementado de forma "relativamente" eficiente

TIPOS DE LÓGICAS

■ Lógica de proposiciones

- Lógica más sencilla
- Representa únicamente hechos discretos del mundo real que pueden ser ciertos o falsos
 - en el mundo representado sólo existen hechos individuales
 - proposición = afirmación (cierta o falsa) sobre un hecho único del dominio
 - pueden combinarse proposiciones (hechos) mediante conectores para expresar hechos más complejos
- Poco poder expresivo
 - Difícil representar conjuntos de hechos con características comunes (generalización)
 - ◊ una proposición por cada hecho
 - ◊ no permite cuantificación ni variables
 - Difícil expresar y generalizar relaciones entre hechos

■ Lógica de predicados de 1^{er} orden

- Más potente y con mayor utilidad práctica
- Representa el mundo en términos de objetos y predicados entre esos objetos
 - predicados representan propiedades de los objetos o relaciones entre objetos
- Permite uso de cuantificadores (\forall , \exists)

3.2.1 Lógica de Proposiciones

(a) SINTAXIS

ELEMENTOS

- Átomos
 - constantes lógicas: *verdadero*(*V*), *falso*(*F*)
 - proposiciones lógicas: (strings de caracteres)
 - ejemplos: $\left\{ \begin{array}{l} p, q, r, \dots \\ encima_mesa_libro \\ juan_es_mortal \end{array} \right.$
- Conectivas lógicas
 - " \wedge " (Y): conjunción lógica
 - $p \wedge q, juan_es_hombre \wedge juan_es_mortal$
 - " \vee " (O): disyunción lógica
 - $p \vee q, juan_trabaja_banco \vee juan_estudia_informatica$
 - " \rightarrow " (**implica**): implicación
 - **No** significa **relación causal** [no "*antecedente implica consecuente*"]
 - $p \rightarrow q, juan_trabaja_banco \rightarrow juan_gana_pasta$
 - " \leftrightarrow " (**equivalente**): equivalencia o bicondicional
 - $p \leftrightarrow q, juan_trabaja_banco \leftrightarrow juan_gana_pasta$
 - " \neg " (**no**): negación
 - $\neg p, \neg juan_trabaja_banco$

FÓRMULAS BIEN FORMADAS (*f.b.f.*) (u oraciones lógicas)

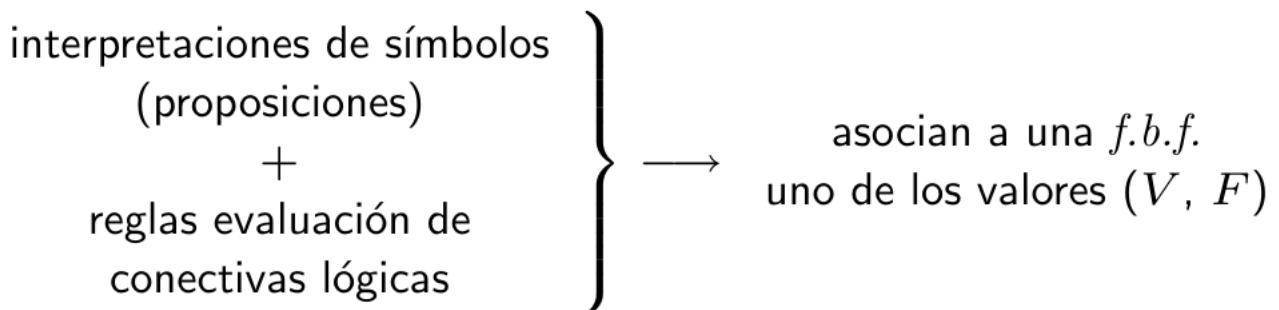
- Cualquier átomo (constantes + proposiciones) es una *f.b.f.*
- Si *F* y *G* son *f.b.f.*, también lo son:
 - $F \vee G$
 - $F \rightarrow G$
 - $\neg F$
 - $F \wedge G$
 - $F \leftrightarrow G$
 - (F)

(b) SEMÁNTICA

En todas las lógicas la semántica de una oración depende de la interpretación concreta que se le asigne a los símbolos (proposiciones / predicados + conectivas), [semántica *composicional*]

Interpretación: Correspondencia entre elementos del lenguaje (símbolos) y los elementos del mundo real que representan

- Los demás elementos del lenguaje (conectivas, cuantificadores) tienen siempre el mismo significado, no dependen del mundo representado
- Una *f.b.f.* (oración) es verdadera o falsa según los valores de verdad o falsedad de los símbolos que la componen



En lógica proposicional → **semántica directa**

- Cada **proposición** se asocia con un **hecho individual** del dominio



- El **valor de verdad** de una proposición depende únicamente de si el hecho que tiene asociado es cierto o falso en el mundo real
 - en lógica de predicados esa asignación no es directa

- En lógica proposicional

Interpretación \equiv asignación valores de verdad a proposiciones

- interpretación \approx entrada en tabla de verdad (sólo en l. proposiciones)

- Dada una interpretación a las proposiciones de una *f.b.f.*, el valor de verdad de esa *f.b.f.* bajo dicha interpretación, se obtiene por decomposición de las conectivas lógicas

p	q	$q \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	F	F	V	V
F	V	F	V	V	F
V	F	F	V	F	F
V	V	V	V	V	V

(*)

(*) No representa relación causal

" \rightarrow " representa: "cuando p es V , también q es V "

$p \rightarrow q$ equivale a $\neg p \vee q$

$$p \rightarrow q \equiv \neg p \vee q$$

- CONCLUSIÓN:

Para probar la verdad/falsedad de una *f.b.f.* en lógica proposicional "bastaría" con construir su tabla de verdad y consultar el valor de verdad para la interpretación considerada

Problema: explosión combinatoria

NOTA: Leyes lógicas

- Pares de *f.b.f.* equivalentes (sus valores de verdad son idénticos para cualquier interpretación)
- Se derivan de las tablas de verdad y de la semántica de las conectivas lógicas

- Ley Asociativa:
$$\left\{ \begin{array}{l} \Psi_1 \vee (\Psi_2 \vee \Psi_3) \equiv (\Psi_1 \vee \Psi_2) \vee \Psi_3 \\ \Psi_1 \wedge (\Psi_2 \wedge \Psi_3) \equiv (\Psi_1 \wedge \Psi_2) \wedge \Psi_3 \end{array} \right.$$
- Ley Conmutativa:
$$\left\{ \begin{array}{l} \Psi_1 \vee \Psi_2 \equiv \Psi_2 \vee \Psi_1 \\ \Psi_1 \wedge \Psi_2 \equiv \Psi_2 \wedge \Psi_1 \end{array} \right.$$
- Ley Distributiva:
$$\left\{ \begin{array}{l} \Psi_1 \wedge (\Psi_2 \vee \Psi_3) \equiv (\Psi_1 \wedge \Psi_2) \vee (\Psi_1 \wedge \Psi_3) \\ \Psi_1 \vee (\Psi_2 \wedge \Psi_3) \equiv (\Psi_1 \vee \Psi_2) \wedge (\Psi_1 \vee \Psi_3) \end{array} \right.$$
- Ley de DeMorgan:
$$\left\{ \begin{array}{l} \neg(\Psi_1 \vee \Psi_2) \equiv \neg\Psi_1 \wedge \neg\Psi_2 \\ \neg(\Psi_1 \wedge \Psi_2) \equiv \neg\Psi_1 \vee \neg\Psi_2 \end{array} \right.$$
- Ley de Idempotencia: $\Psi_1 \vee \Psi_1 \equiv \Psi_1 \equiv \Psi_1 \wedge \Psi_1$
- Ley de Medio Excluido: $\neg\Psi \vee \Psi \equiv V, \neg\Psi \wedge \Psi \equiv F$
- Ley de Elemento neutro: $\Psi \vee F \equiv \Psi, \Psi \wedge V \equiv \Psi$
- Ley de Elemento nulo: $\Psi \vee V \equiv V, \Psi \wedge F \equiv F$
- Ley de Contraposición: $\Psi_1 \rightarrow \Psi_2 \equiv \neg\Psi_1 \vee \Psi_2$

SATISFACTIBILIDAD Y MODELOS

- Una interpretación (=asignación de verdad) I que hace V a una f.b.f. Ψ se dice que es un **modelo** de Ψ

- Ejemplo: $\left\{ \begin{array}{l} p : \text{"juan es alto"} \\ q : \text{"luis estudia informática"} \\ r : \text{"juan estudia informática"} \end{array} \right.$

$$f.b.f. \Psi = p \wedge (q \vee \neg r)$$

p	q	r	$q \vee \neg r$	$\Psi = p \wedge (q \vee \neg r)$
F	F	F	V	F
F	F	V	F	F
F	V	F	V	F
F	V	V	V	F
(*)	V	F	F	V
	V	F	F	F
(*)	V	V	F	V
(*)	V	V	V	V

(*) modelos de Ψ : de todas las interpretaciones posibles, Ψ sólo es cierta en estas
 (limita los mundos donde la f.b.f. es posible)
 mundos posibles vs. mundos donde Ψ en V
 (interpretaciones) (modelos)

- Dado un conjunto de f.b.f. $\Phi = \{\Psi_1, \Psi_2, \dots, \Psi_n\}$, una interpretación I es un **modelo del conjunto** Φ si hace que sean V todas sus f.b.f..
 - Es decir, I es un modelo de " $\Psi_1 \wedge \Psi_2 \wedge \dots \wedge \Psi_n$ "
- El conjunto de f.b.f. Φ "restringe" los mundos posibles
 $Phi \equiv$ base de conocimiento (B.C.)

- Una *f.b.f.* Ψ que tiene al menos un modelo, se dice que es **satisfacible**
 - Ψ es **insatisfacible** si no tiene modelos
- Una *f.b.f.* Ψ que es cierta para todas las posibles interpretaciones se dice que es **válida** (también llamada **tautología**) [verdad "sintáctica"]
 - Notación: $\models \Psi$

Ψ es V independientemente de la interpretación considerada

 - Siempre es V como consecuencia de su estructura sintáctica
 - *Ejemplo:* " $p \vee \neg p$ " siempre es V
- Dos *f.b.f.* Ψ_1 y Ψ_2 son **equivalentes** si sus valores de verdad son iguales bajo cualquier interpretación
 - Es decir, " $\Psi_1 \leftrightarrow \Psi_2$ " es una tautología
 - Notación: $\Psi_1 \equiv \Psi_2$

CONSECUENCIA LÓGICA

- Siendo $\Phi = \{\Psi_1, \dots, \Psi_n\}$ un conjunto de f.b.f.. Si Ψ tiene valor V para todas las interpretaciones para las cuales cada f.b.f. contenida en Φ es V , diremos que Ψ es **consecuencia lógica** de Φ .
 - Otra definición: " Ψ es consecuencia lógica de Φ si todo modelo de Φ es también un modelo de Ψ "
 - Tambien: $\begin{cases} \Phi \text{ "implica lógicamente a" } \Psi \\ \Psi \text{ "se sigue de" } \Phi \end{cases}$
 - Notación: $\boxed{\Phi \models \Psi}$
 - Ejemplo: $\{p\} \models p$, $\{p, p \wedge q\} \models q$, $\{p, p \rightarrow q\} \models q$

p	q	$p \wedge (p \rightarrow q)$	q
F	F	F	F
F	V	F	V
V	F	F	F
(*) V	V	V	V

(*): modelo de $\Phi = \{p, p \rightarrow q\}$ y de q
 RESULTADO: " $\{p, p \rightarrow q\} \models q$ "

- En nuestro caso (I.A.), Φ se corresponderá con la **Base de Conocimientos (B.C.)** y Ψ con una hipótesis/conclusión
 - Recopila todo lo que sabemos de un dominio en forma de f.b.f. (conocimiento de partida)
 - Será un conocimiento (conj. de f.b.f.) que dentro de nuestro dominio supondremos siempre verdadero

Φ restringe las interpretaciones (mundos posibles) donde Ψ se deberá verificar

$\Phi \models \Psi \equiv$ cuando se verifique Φ , también lo hará Ψ

Φ : base de conocimiento (conocimiento de partida)

Ψ : hipótesis/conclusión

(c) REGLAS DE INFERENCIA

Mecanismos **sintácticos** que permiten deducir *f.b.f.* a partir de otras *f.b.f.* distintas (no tiene en cuenta significado)

- Representan patrones de inferencias usados comúnmente
- Procedimientos formales: nuevas *f.b.f.* resultan de **transformar sintácticamente** la forma de las *f.b.f.* de partida
- No se tiene en cuenta el significado de los elementos de la lógica (proposiciones en este caso)

Reglas más comunes (preservan la "consecuencia lógica")

$$1. \text{ Modus Ponens: } \frac{\Psi_1}{\Psi_1 \rightarrow \Psi_2}$$

$$2. \text{ Modus Tollens: } \frac{\neg\Psi_2}{\Psi_1 \rightarrow \Psi_2}$$

$$3. \text{ Introducción de la Conjunción: } \frac{\Psi_1 \quad \Psi_2}{\Psi_1 \wedge \Psi_2}$$

$$4. \text{ Introducción de la Disyunción: } \frac{\Psi_1}{\Psi_1 \vee \Psi_2} \quad \text{ó} \quad \frac{\Psi_2}{\Psi_1 \vee \Psi_2}$$

$$5. \text{ Eliminación de la Conjunción: } \frac{\Psi_1 \wedge \Psi_2}{\Psi_1} \quad \text{ó} \quad \frac{\Psi_1 \wedge \Psi_2}{\Psi_2}$$

NOTACIÓN: Si la *f.b.f.* Ψ se obtiene del conjunto de fórmulas Φ , aplicando la regla de inferencia R se notará:

Notación: $\boxed{\Phi \vdash_R \Psi}$

DEMOSTRACIÓN

- Dado un conjunto de f.b.f. Φ se dirá que una secuencia de f.b.f. $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ es una **demonstración** de la f.b.f α_n a partir de Φ empleando un conjunto de reglas de inferencia R si y sólo si:

$$\forall \alpha_i \left\{ \begin{array}{l} \bullet \text{ o bien } \alpha_i \in \Phi \\ \bullet \text{ o bien } \alpha_i \text{ puede ser inferida} \\ \quad \text{de } \Phi \cup \{\alpha_1, \alpha_2, \dots, \alpha_{i-1}\} \\ \quad \text{aplicando una de las reglas del} \\ \quad \text{conjunto } R \end{array} \right.$$

Demostración \approx conjunto ordenado de reescrituras

- NOTACIÓN: $\boxed{\Phi \vdash_R \alpha_n}$ = '' α_n se obtiene de Φ mediante R ''
- Si existe una demostr. de α_n , se dice que α_n es un **teorema** de Φ
- Ejemplo: $\left\{ \begin{array}{l} p : \text{"Juan estudia"} \\ q : \text{"Luis aprueba"} \\ r : \text{"Juan es rubio"} \end{array} \right.$

Conocim. inicial: $\Phi = \{p, r, p \rightarrow q\}$

Aplicando reglas $R = \{\text{modus ponens}, \text{introd. conjunc}\}$
se demuestra (*construye*) " $\Phi \vdash q \wedge r$ "

$$\begin{aligned} (a) : \quad & \Phi = \{p, r, p \rightarrow q\} \vdash_{\text{modus ponens}} q \\ (a) : \quad & \Phi \cup \{q\} = \{p, r, p \rightarrow q, q\} \vdash_{\text{introd. conj.}} q \wedge r \end{aligned}$$

Representación

METATEOREMAS

Teoremas a cerca del cálculo proposicional y de predicados (no son parte de la lógica de proposiciones/predicados)

1. Deducción:

- "Si $\Phi = \{\Psi_1, \Psi_2, \dots, \Psi_n\} \models \Psi$ entonces

$\Psi_1 \wedge \Psi_2 \wedge \dots \wedge \Psi_n \rightarrow \Psi$ es **válido** (tautología)"

- a) Por def. de consecuencia lógica (\models), antecedente y consecuente tendrán los mismos modelos (son V a la vez)
 - b) Por tabla de verdad de la conectiva " \rightarrow ", el resultado del *implica* será siempre V
- Leido en el otro sentido:
 - "Si $\Psi_1 \wedge \Psi_2 \wedge \dots \wedge \Psi_n \rightarrow \Psi$ es válido, entonces $\Phi \models \Psi$ "

2. Reducción al absurdo:

- "Si Φ tiene un modelo y $\Phi \cup \{\neg\Psi\}$ no lo tiene, entonces $\Phi \models \Psi$ "
 - Ψ no se contradice con el conocim. de Φ .

Utilidad: proporcionan criterios para determinar si se cumple o no la relación de *consecuencia lógica* $\Phi \models \Psi$.

COMPLETITUD Y SOLIDEZ (propiedades reglas de inferencia)

1. "Una regla de inferencia R es **sólida** si siempre que $\Phi \vdash_R \Psi$, entonces $\Phi \models \Psi$ "

- Sólido \approx Correcto (SOUND)
- La regla sólo genera/demuestra f.b.f. que son consecuencia lógica de Φ (no construye f.b.f falsas)
- Genera oraciones que siempre son verdaderas en el dominio actual (definido por Φ)

- *Ejemplo:* *modus ponens* es sólido (correcto):
$$\frac{\alpha \rightarrow \beta \quad \alpha}{\beta}$$

Tenemos que, usando *modus ponens*, $\{\alpha \rightarrow \beta, \alpha\} \vdash \beta$.

¿es cierto que $\{\alpha \rightarrow \beta, \alpha\} \models \beta$?

Si, porque " $(\alpha \wedge (\alpha \rightarrow \beta)) \rightarrow \beta$ " es una tautología (metateorema deducción)

α	β	$\alpha \wedge (\rightarrow \beta)$	$(\alpha \wedge (\alpha \rightarrow \beta)) \rightarrow \beta$
F	F	F	V
F	V	F	V
V	F	F	V
(*)	V	V	V

2. "Si siempre que $\Phi \models \Psi$ existe una demostración que usando R permite que $\Phi \vdash_R \Psi$, se dirá que R es **completa**"

- La regla R es capaz de proporcionar una demostración a todas las f.b.f. que son consecuencia lógica de Φ .

EN RESUMEN:

Si contáramos con una regla R que fuera sólida y completa podríamos determinar todas las f.b.f. que son consecuencia lógica de Φ (base de conocimiento actual), mediante la búsqueda de una demostración.

- Tendríamos un método que podría generar todo el conocimiento cierto derivable del conocimiento de partida.

RESOLUCIÓN Y PRUEBA POR REFUTACIÓN

- **Resolución:** Regla de inferencia (para lóg. propos. y lóg. predicados)

$$\frac{\alpha \vee \beta \quad \neg\alpha \vee \gamma}{\beta \vee \gamma} \quad \left. \begin{array}{l} \alpha \vee \beta \\ \neg\alpha \vee \gamma \\ \beta \vee \gamma \end{array} \right\} : \begin{array}{l} \text{claúsulas} \\ \alpha : \text{átomo resuelto} \\ \beta \vee \gamma : \text{claúsula resolvente} \end{array}$$

- Es la más potente: todas las demás reglas de inferencia son casos particulares de resolución
 - Ej.: *modus ponens* caso especial de resolución

$$\frac{\alpha \quad \alpha \rightarrow \beta}{\beta} \quad \rightsquigarrow \quad \frac{\alpha \vee Falso \quad \neg\alpha \vee \beta}{Falso \vee \beta}$$

- Puede usarse como única regla de inferencia en sistemas de deducción automática
- Exige que conocimiento esté representado de "forma adecuada"
 - Regla de resolución sólo maneja claúsulas
 - Conversión a *forma normal conjuntiva*(FNC)

1. Regla de resolución es **sólida**

- Se cumple: " $\Phi \vdash_{\text{resolución}} \Psi$ implica $\Phi \models \Psi$ "

$$\frac{\alpha \vee \beta \quad \neg \alpha \vee \gamma}{\beta \vee \gamma} \quad \{ \alpha \vee \beta, \neg \alpha \vee \gamma \} \vdash \beta \vee \gamma$$
$$(\Phi \vdash \Psi)$$

- ¿ $\Phi \models \Psi$? \rightarrow tenemos que demostrar que cuando Φ es *V*, también lo es Ψ .

(los modelos de Φ son modelos de Ψ)

- **Razonamiento por casos**

- Para un modelo de Φ : " $\alpha \vee \beta$ es *V*" y " $\neg \alpha \vee \gamma$ es *V*".
- Si en ese modelo α es *V*, $\neg \alpha$ será *F* y, para que $\neg \alpha \vee \gamma$ sea *V*, γ debe ser *V*.
- En caso contrario, al ser α *F*, β debe ser *V*.
- Por lo tanto, en cualquier caso, $\beta \vee \gamma$ será *V*.

2. Regla de resolución **no** es **completa**

- No se cumple: " $\Phi \models \Psi$ implica $\Phi \vdash_{\text{resolución}} \Psi$ "
- Existen implicaciones lógicas que no se pueden demostrar/construir usando resolución
- Contraejemplo:
 - Es cierto que " $\{p, r\} \models p \vee r$ ".
 - Pero: $\{p, r\} \not\models \neg_{\text{resolución}} p \vee r$
- No podemos inferir $p \vee r$ del conjunto $\{p, r\}$ usando resolución
 - No hay posibilidad de resolver ninguna *f.b.f.*
- CONCLUSIÓN: No podemos usar resolución para demostrar **todas** las relaciones de consecuencia lógica

PRUEBA POR REFUTACIÓN

- Si se puede usar resolución para demostrar que la negación de una conclusión $\neg\Psi$ es **inconsistente** con el conocimiento del mundo Φ
 - Demostración por *reducción al absurdo*
 - Metateorema: $\Phi \models \Psi$ equivale a " $\Phi \cup \neg\Psi$ es *insatisfacible*"
 - IDEA: Añadir a Φ la negación de la conclusión Ψ y aplicar resolución hasta que pase "algo"
 - Si se puede llegar a *construir* una cláusula vacía (*Falso*), hay una **contradicción** y, por lo tanto, se verifica $\Phi \models \Psi$
 - Si no se llega a una cláusula vacía, Φ y $\neg\Psi$ son consistentes y $\Phi \not\models \Psi$

CONVERSIÓN A FORMA CLAUSAL

- Paso previo a la aplicación de resolución
- Resolución maneja *f.b.f.* en forma de disyunciones de átomo, llamadas **cláusulas**
- Es necesario transformar todas la *f.b.f.* a una forma normalizada

f.b.f. \rightsquigarrow conjunciones de disyunciones (cláusulas) (*FNC*)

- *FNC*: forma normal conjuntiva (conjunción de disyunciones)
- Otras formas normales:
 - *FND*: forma normal disyuntiva (disyunción de conjunciones)
 - *FNI*: forma normal implicativa (conjunción de implicaciones)

■ Conversión a FNC

- Cualquier *f.b.f.* se puede convertir a FNC

- **Pasos:**

1. Eliminar implicaciones

- Usar relaciones $\begin{cases} p \rightarrow q \equiv \neg p \vee q \\ p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p) \end{cases}$

2. Reducir ámbito de las negaciones

- \neg debe aplicarse sólo sobre átomos

- Usar DeMorgan + doble neg.: $\begin{cases} \neg(p \vee q) \equiv \neg p \wedge \neg q \\ \neg(p \wedge q) \equiv \neg p \vee \neg q \\ \neg\neg p \equiv p \end{cases}$

3. Reordenar *f.b.f.* para obtener FNC

- Aplicar leyes distributiva $\begin{cases} \text{distribuir } \wedge \text{ sobre } \vee \\ \text{distribuir } \vee \text{ sobre } \wedge \end{cases}$ y asociativa

$$a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$$

$$a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$$

- *Ejemplo:* Pasar $\neg(p \rightarrow q) \vee (r \rightarrow p)$ a FNC

$$1 : \quad \neg(p \rightarrow q) \vee (r \rightarrow p) = \neg(\neg p \vee q) \vee (\neg r \vee p) =$$

$$2 : \quad = (\neg\neg p \wedge \neg q) \vee (\neg r \vee p) =$$

$$= (p \wedge \neg q) \vee (\neg r \vee p) =$$

$$3 : \quad = (p \vee \neg r \vee p) \wedge (\neg q \vee \neg r \vee p)$$

Claúsulas resultantes: $\{\neg r \vee p, \neg q \vee \neg r \vee p\}$

PASOS DEMOSTRACIÓN POR REFUTACIÓN

1. Convertir las *f.b.f.* de Φ a FNC
2. Negar la *f.b.f.* Ψ que deseamos probar y convertirla a FNC
→ pasar $\neg\Psi$ a FNC
3. Unir los conjuntos de cláusulas obtenidas para Φ y $\neg\Psi$ en un conjunto Π .
4. Aplicar, de forma exhaustiva, resolución sobre pares de cláusulas de Π , añadiendo los resolventes a Π
 - a) Seleccionar un par de cláusulas de Π con los átomos a resolver adecuados
 - b) Añadir el resultado de la resolución al nuevo conjunto Π
5. Parar cuando:
 - Se genera una **cláusula vacía** ($=\text{False}$) (Hay contradicción)
→ Se verifica: $\Phi \models \Psi$
 - No haya más resolventes nuevos (No hay contradicción)
→ Se verifica: $\Phi \not\models \Psi$

Ejemplo: Demostrar $\{p, r\} \models p \vee r$

- $\Phi = \{p, r\}$ ya está en FNC
- Negar $\Psi = p \vee r$ y pasar a FNC

$$\neg\Psi = \neg p \wedge \neg r \quad (\text{ya está en FNC})$$

- Unir: $\Pi = \{p, r, \neg p, \neg r\}$
- Aplicar todas las resoluciones posibles

$$\frac{p, \neg p}{\{\}} \qquad \frac{r, \neg r}{\{\}}$$

Ejemplo: Un robot puede mover un objeto si tiene la batería encendida y el objeto es desplazable

- Base de conocimiento: (conocimiento actual del mundo)

$$\Phi = \{ \quad \text{bateria_encendida}, \\ \neg \text{robot_se_mueve} \\ \text{bateria_encendida} \wedge \text{objeto_desplazable} \rightarrow \text{robot_se_mueve} \}$$

- Queremos saber si en la situación actual el objeto no es desplazable

$$\Psi = \neg \text{objeto_desplazable}$$

1. Pasar Φ a FNC

$$\Phi = \{ \quad \text{bateria_encendida}, \\ \neg \text{robot_se_mueve} \\ \neg \text{bateria_encendida} \vee \neg \text{objeto_desplazable} \vee \text{robot_se_mueve} \}$$

2. Negar Ψ y pasar a FNC

$$\neg P_{\text{si}} = \text{objeto_desplazable}$$

3. Unir Φ y $\neg \Psi$

$$\Pi = \{ \quad \text{bateria_encendida}, \\ \neg \text{robot_se_mueve} \\ \neg \text{bateria_encendida} \vee \neg \text{objeto_desplazable} \vee \text{robot_se_mueve}, \\ \text{objeto_desplazable} \}$$

4. Aplicar resolución de forma exhaustiva hasta encontrar cláusula vacía o agotar posibilidades (ver esquema)

5. Alcanza cláusula vacía \Rightarrow Hemos demostrado que “ $\Phi \models \Psi$ ”

PROPIEDADES REFUTACIÓN POR RESOLUCIÓN EN LÓGICA DE PROPOSICIONES

- Refutación mediante resolución es **sólida** en lógica de proposiciones
 - "Si partiendo de Φ y $\neg\Psi$, el procedimiento de refutación por resolución genera la cláusula vacía, entonces $\Phi \models \Psi$,"
- Refutación mediante resolución es **completa** en lógica de proposiciones
 - "Si $\Phi \models \Psi$, el procedimiento de refutación por resolución generará la cláusula vacía"
- Refutación mediante resolución es **decidible** en lógica de proposiciones
 - "Si Φ es finito y $\Phi \not\models \Psi$, el procedimiento de refutación por resolución terminará sin haber generado la cláusula vacía"
→ el nº de resoluciones posibles a comprobar es finito

ESTRATEGIAS DE RESOLUCIÓN

- En el procedimiento de refutación por resolución:
 1. Es necesario un proceso de búsqueda para identificar las cláusulas a resolver en cada iteración
 2. Es necesario, además, establecer un "orden" en la aplicación de las posibles resoluciones
- Problema análogo a la expansión de nodos en las búsquedas en espacios de estados
- Las estrategias de control $\left\{ \begin{array}{l} \text{elección de los átomos} \\ \text{elección del par de cláusulas a resolver} \end{array} \right.$ determinan la eficiencia del procedimiento de refutación
→ Facilitan/dificultan encontrar contradicciones (cláusulas vacías)

- Árboles de resolución:
 - Conjunto de árboles inversos
 - *Hojas*: cláusulas del conjunto inicial Π
 - *Nodos internos*: resolventes obtenidos de las hojas o de otros resolventes, aplicando resolución
 - *Raíz/es*: resultado de la refutación
- *Definiciones*:
 - Cláusulas inciales = resolventes de nivel 0
 - Un resolvente de nivel $i + 1$ se obtiene de un resolvente de nivel i y otro de nivel $j \leq i$

Estrategias:

1. Estrategias generales:
 - Estrategia **1º en anchura**: genera todos los resolventes de nivel k antes de obtener resolventes de nivel $k + 1$
 - Estrategia **1º en profundidad**: busca generar siempre que sea posible resolventes del nivel siguiente al actual
2. Preferencia por la unidad
 - Se efectúan en primer lugar las resoluciones donde, al menos, una de las dos cláusulas implicadas esté formada por un único átomo (*cláusulas unitarias*)
 - Siempre se genera una nueva cláusula más pequeña que la no unitaria de partida
 - *Ejemplo*: En el ejemplo del robot todas las resoluciones tienen al menos una cláusula unitaria

3. Conjunto soporte

- DEF.: **Conjunto soporte:** Conjunto formado por las cláusulas resultantes de la negación de la hipótesis (Ψ) que se quiere demostrar y por todas las cláusulas que "descienden" de ellas como resultado de las resoluciones realizadas hasta el momento
- IDEA BASE: Realizar únicamente resoluciones donde intervenga, al menos, una cláusula del conjunto soporte
 - Nueva cláusula generada pasará a formar parte del conj. soporte
- Intuitivamente, las cláusulas del conjunto soporte son las que "parecen relevantes" respecto a la demostración de la cláusula objetivo Ψ
- VENTAJAS:
 - Se reduce el espacio de búsqueda:
 - No se buscan 2 cláusulas en todo el conjunto Π
 - Toma 1 del conj. soporte y 1 de Π que empareje con ella
 - Es completa:
 - Se puede demostrar que usando al menos una cláusula del conjunto soporte en cada resolución, se asegura que se va a encontrar una contradicción (cláusula vacía), siempre que ésta exista.
- *Ejemplo:* En el ejemplo del robot todas las resoluciones tienen una cláusula del conj. soporte.
 - Empieza usando la hipótesis negada ($\neg\Psi$) y en cada paso resuelve una de sus descendientes con otra cláusula de Π .

4. Resolución lineal

- Sólo considera resoluciones donde, al menos, una de las 2 cláusulas a resolver pertenezca al conjunto de cláusulas inicial ($\Pi = \Phi \cup \{\neg\Psi\}$)
 - Es decir, en todas las resoluciones interviene una cláusula incial (nivel 0)
- *Ejemplo:* En el ejemplo del robot todas las resoluciones tienen al menos una cláusula inicial.
- **No es completa** (puede no encontrar la contradicción, aunque la haya)
- Usada en intérpretes Prolog sobre cláusulas de Horn