T3. Lógica proposicional

Lecturas :

- CAPÍTULOS 13,14 de Nilsson
- CAPÍTULO 7 de Russell & Norvig
- Gödel, Escher, Bach: An eternal Golden Braid Douglas R. Hofstadter

Introducción

⌘ **Representaciones del mundo**

- **Icónicas (por analogía):** simulación de los aspectos relevantes del entorno y de los efectos de las acciones de los agentes sobre él.
- **Basadas en características:** Vector de valores de características (atributos) + descripciones declarativas del entorno.
  - **Leyes generales**: "Los objetos redondos no tienen bordes".
  - **Información negativa**: "No estamos en Tokio"
  - **Información con incertidumbre**: "Estamos en Amberes o en Dublín".
  - **Restricciones** sobre los valores de las variables: "Las notas en este curso están entre 0 y 10".

⌘ **Razonamiento:** Los valores de ciertas características pueden ser inferidos a partir de los valores de otras características + conocimiento del problema.

<u>Ejemplo</u>: Robot dotado de un brazo mecánico que intenta levantar un objeto

Características x_1(BATERÍA_OK), x_2 (BRAZO_ACCIONADO), x_3 (ALZABLE), x_4 (SE_MUEVE)

El robot detecta directamente BATERÍA_OK, BRAZO_ACCIONADO, SE_MUEVE, pero no ALZABLE

Conocimiento: BATERÍA_OK $\wedge$ BRAZO_ACCIONADO $\wedge$ ALZABLE $\Rightarrow$ SE_MUEVE.

El robot lee {BATERÍA_OK (1), BRAZO_ACCIONADO (1), SE_MUEVE (0)}

e infiere ALZABLE (0)

Nuestro objetivo: Un agente basado en conocimiento

<u>Base de conocimiento:</u> Conjunto de sentencias en un lenguaje formal, cada una representando una aserción acerca del mundo.

- El agente puede representar conocimiento acerca del mundo incluyendo **átomos** en su **base de conocimiento (BC)**. La ocurrencia explícita de un átomo en la base de conocimiento del agente significa que el agente considera ***Verdadera*** en su mundo la proposición asociada.

⌘ **Un agente basado en conocimiento debería tener:**

- **Mecanismos en los que basar su conocimiento:**
  - **Conocimiento 'innato' de partida.**
  - Un **sistema sensorial** para determinar **directamente** la **verdad o falsedad de las proposiciones** acerca de su **mundo**.
- **Un mecanismo de consulta** que permita acceder al conocimiento almacenado en la BC.
- **Un mecanismo de inferencia correcto** que permita deducir nuevas sentencias y añadirlas a su base de conocimiento: Si las fbfs de la BC tienen valor de verdad *Verdadero* en el mundo real, entonces las fbfs derivadas de la BC mediante reglas de inferencia correctas tienen valor de verdad *Verdadero* en el mundo real.
- Es también deseable la **completitud** del mecanismo de inferencia.

  (Gödel dice: esto no es siempre posible)

Los elementos de la lógica

- **Un lenguaje formal**: Símbolos + reglas sintácticas para combinar símbolos en sentencias gramáticas.
- **Reglas de inferencia**: Reglas tipográficas para construir nuevas sentencias partiendo de un grupo dado de sentencias. (*Tipográficas*: no usan semántica, es decir, no se meten en el significado de las sentencias)
- **Semántica**: Reglas de interpretación que asocian sentencias en el lenguaje con afirmaciones en el dominio de discurso.

Olvida el significado.

Considera a la lógica como un juego con reglas bien definidas.

Para poder jugar hay que conocer y aplicar correctamente dichas reglas.

PRECAUCIÓN: Cuando se está haciendo un ejercicio de deducción lógica formal es fácil (y a menudo lleva a conclusiones erróneas) hacer trampa y hacer deducciones informales basadas en el significado.

Lenguaje, I

- **Átomos**: V , F
  P, Q, P1, P2, P3, ..., Pn, ...

- **Paréntesis:** (,)

- **Conectivas proposicionales**:
  $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ (en orden de prioridad)

- **Fórmulas bien formadas** (fbfs o **sentencias)**
  - **Un átomo es una fbf.**
  - **Si** w_1 **y** w_2 son fbfs, entonces también lo son las expresiones:

    $\neg w_1$ (negación de w_1).

    $(w_1 \wedge w_2)$ (conjunción de w_1 y w_2).

    $(w_1 \vee w_2)$ (disyunción de w_1 y w_2).

    $(w_1 \Rightarrow w_2)$ (Implicación, condicional, regla:
    w_1 es la premisa o antecedente;
    w_2 es la conclusión, o consecuente).

    $(w_1 \Leftrightarrow w_2)$ (bicondicional, o afirmación si-y-sólo-si)

Lenguaje, II

- **Gramática (en forma de Backus-Naur)**
  - Sentencia $\rightarrow$ Sentencia Atómica | Sentencia Compleja
  - Sentencia Atómica $\rightarrow$ V | F | Símbolo
  - Símbolo $\rightarrow$ P | Q | P1 | P2 | P3 | ... | Pn, ...
  - Sentencia Compleja $\rightarrow$ $\neg$Sentencia
    | (Sentencia $\wedge$ Sentencia)
    | (Sentencia $\vee$ Sentencia)
    | (Sentencia $\Rightarrow$ Sentencia)
    | (Sentencia $\Leftrightarrow$ Sentencia)

- **Nota:** Esta gramática es muy estricta con los paréntesis.

  En las notas que siguen se omiten algunos paréntesis y se usan las reglas de prioridad en las conectivas. Los paréntesis (,) se usarán sólo para alterar la prioridad o para mejorar la legibilidad de las fórmulas. Serán considerados como separadores extra-lingüísticos que agrupan fbfs en sub-fbfs.

- **Nota:** En USA se usa $\supset$ en lugar de $\Rightarrow$

Ejemplos: $((P\wedge Q)\Rightarrow\neg P)$, o $P\wedge Q\Rightarrow\neg P$

$(P\Rightarrow\neg P)$, o $P\Rightarrow\neg P$

$((P\vee P)\Rightarrow\neg P)$, o $P\vee P\Rightarrow\neg P$

Reglas de inferencia

- **Reglas de inferencia:** Reglas tipográficas que especifican cómo generar nuevas fbfs a partir de un conjunto dado de fbfs

- Un posible conjunto **R** de **reglas de inferencia es**:

Sean w_1, w_2 fbfs

(1) ***Modus ponens*** :	$\{w_1, w_1 \Rightarrow w_2\} \vdash_{M.P.} w_2$
(2) ***Modus tollens***:	$\{\neg w_2, w_1 \Rightarrow w_2\} \vdash_{M.T.} \neg w_1$
(3) Introducción $\wedge$:	$\{w_1, w_2\} \vdash_{\wedge\ INTRO} w_1 \wedge w_2$
(4) Conmutatividad de $\wedge$:	$\{w_1 \wedge w_2\} \vdash_{\wedge\ COMMUTA} w_2 \wedge w_1$
(5) Eliminación $\wedge$:	$\{w_1 \wedge w_2\} \vdash_{\wedge\ ELIMIN} w_1$
(6) Introducción $\vee$**:**	$\{w_1\} \vdash_{\vee\ INTRO} w_1 \vee w_2$
	$\{w_2\} \vdash_{\vee\ INTRO} w_1 \vee w_2$
(7) Eliminación $\neg\neg$:	$\{\neg\neg w_1\} \vdash_{\neg\neg ELIMIN} w_1$

Este conjunto **R** de reglas de inferencia es **correcto** pero **no completo** ("correcto" y "completo" se definirán más adelante)

Pruebas y teoremas

- **Prueba:** La secuencia de fbfs

  $\{w_1, w_2, ..., w_{n-1}, w_n\}$

  es una prueba (o deducción, o derivación) de w_n a partir de un conjunto de fbfs Δ con el conjunto de reglas de inferencia *R* sii (si y sólo si) cada $w_i, i=1,2,...,n$ está o bien en Δ o puede ser deducido de $\{w_1, w_2, ..., w_{i-1}\}$ por la aplicación de alguna regla de inferencia de *R.*

- **Teorema**:

  Si existe una prueba de w_n a partir de Δ con las reglas de inferencia *R* , entonces w_n es un **teorema** de Δ con reglas de inferencia *R*

$$\Delta \vdash_R w_n$$

Semántica: Tablas de verdad

- **Tablas de verdad:** Reglas que especifican el **valor de verdad de una fbf a partir de los valores de los átomos** que forman la fbf.

  - $\neg w_1$ tiene valor *Verdadero* () si w_1 tiene valor *Falso.*
    tiene valor *Falso* si w_1 tiene valor *Verdadero.*
  - $w_1 \wedge w_2$ tiene valor *Verdadero* sólo si w_1 y w_2 tienen valor *Verdadero*;
    en otro caso, su valor es *Falso.*
  - $w_1 \vee w_2$ tiene valor *Verdadero* si o bien w_1 o w_2 tienen valor *Verdadero*;
    en otro caso, su valor es *Falso.*
  - $w_1 \Rightarrow w_2$ tiene el mismo valor de verdad que $(\neg w_1 \vee w_2)$
  - $w_1 \Leftrightarrow w_2$ tiene el mismo valor de verdad que $(w_1 \Rightarrow w_2) \wedge (w_2 \Rightarrow w_1)$

w_1	w_2	$\neg w_1$	$w_1 \wedge w_2$	$w_1 \vee w_2$	$w_1 \Rightarrow w_2$	$w_1 \Leftrightarrow w_2$
Verdadero	*Verdadero*	*Falso*	*Verdadero*	*Verdadero*	*Verdadero*	*Verdadero*
Verdadero	*Falso*	*Falso*	*Falso*	*Verdadero*	*Falso*	*Falso*
Falso	*Verdadero*	*Verdadero*	*Falso*	*Verdadero*	*Verdadero*	*Falso*
Falso	*Falso*	*Verdadero*	*Falso*	*Falso*	*Verdadero*	*Verdadero*

Trucos

- Las siguientes sustituciones transforman la evaluación de valores de verdad de una fbf en lógica proposicional a un problema aritmético

Lógica	**Aritmética**
Falso	0
Verdadero	1
$\wedge$	$\cdot$
$\vee$	+

- Algunas veces se abrevia $\overline{P}$ por $\neg P$

Semántica: Interpretación

- **Interpretación:**
  - Una interpretación es un conjunto de **asociaciones** entre **átomos y proposiciones acerca del mundo**.
    - En una interpretación, la proposición asociada a un átomo se denomina **denotación de ese átomo**.
  - **Bajo** una **interpretación dada los átomos tienen valores de verdad** (*Verdadero* o *Falso*) que están **determinadas por la verdad o falsedad de la correspondiente proposición** en el mundo.
  - El **átomo** especial **V** tiene siempre el valor ***Verdadero***.
  - El **átomo** especial **F** tiene siempre el valor ***Falso***.

Semántica: Interpretación

- **Semántica (significado de una fbf) ≡ interpretación + aplicación de tablas de verdad**

Deducción

Conjunto de sentencias → Sentencias deducidas

Representación

Semántica ↓ Semántica ↓

Mundo real

Aserciones sobre aspectos del mundo real

Aserciones sobre aspectos del mundo real

Satisfacibilidad

⌘ **Satisfacibilidad:**

- Una **interpretación satisface** un **conjunto de fbfs** Δ si **todas las fbfs** del conjunto tienen valor ***Verdadero*** bajo esa interpretación. Se dice entonces que esa **interpretación** es un **modelo de** Δ.
- Un **conjunto de fbfs** Δ es **satisfacible** si existe al menos una interpretación bajo la cual **todas las fbfs** del conjunto Δ **tienen valor *Verdadero*.**
- La **fbf w** es **satisfacible** si el **conjunto {w} es satisfacible**.

⌘ **Inconsistencia**:

- Un **conjunto de fbfs** se dice **inconsistente** o **insatisfacible** si no hay ninguna interpretación bajo la cual se satisfacen todas las fbfs del conjunto.
- Se dice que una **fbf w** es **insatisfacible** si el **conjunto {w}** es **insatisfacible**.

Ejemplos: F; $P\wedge\neg P$; $\{P\vee Q, P\vee\neg Q, \neg P\vee Q, \neg P\vee\neg Q\}$

⌘ **Validez:** Se dice que una fbf es válida si su valor es *Verdadero* bajo todas las interpretaciones posibles.

Ejemplos: V; $V\vee P$; $\neg P\vee P$; $P\Rightarrow P$; $P\Rightarrow(Q\Rightarrow P)$; $((P\Rightarrow Q)\Rightarrow P)\Rightarrow P$

- Una fbf válida es una **tautología** (no aporta información sobre el mundo).

⌘ **Metateorema 1:** si $\neg w$ es insatisfacible, entonces la fbf w es válida, y viceversa

Equivalencia

Se dice que dos fbfs son lógicamente **equivalentes (≡)** si y sólo si tienen los **mismos valores de verdad bajo todas las interpretaciones posibles**.
Usando las tablas de verdad se pueden obtener las siguientes **propiedades**

- **Absorción:** $(w_1 \wedge V) \equiv w_1$; $(w_1 \vee F) \equiv w_1$
- **Ley del medio excluído:** $(w_1 \vee \neg w_1) \equiv V$
- **Ley de no contradicción:** $(w_1 \wedge \neg w_1) \equiv F$;
- **Leyes de dominación:** $(w_1 \wedge F) \equiv F$; $(w_1 \vee T) \equiv V$
- **Idempotencia:** $(w_1 \wedge w_1) \equiv w_1$; $(w_1 \vee w_1) \equiv w_1$
- **Eliminación de la doble negación:** $\neg\neg w_1 \equiv w_1$
- **Leyes de De Morgan:**

  $\neg(w_1 \vee w_2) \equiv \neg w_1 \wedge \neg w_2$

  $\neg(w_1 \wedge w_2) \equiv \neg w_1 \vee \neg w_2$

Equivalencia

- **Contraposición:** $w_1 \Rightarrow w_2 \ \equiv \ \neg w_2 \Rightarrow \neg w_1$
- **Conmutatividad:**

  $w_1 \vee w_2 \ \equiv \ w_2 \vee w_1$

  $w_1 \wedge w_2 \ \equiv \ w_2 \wedge w_1$
- **Asociatividad:**

  $(w_1 \wedge w_2) \wedge w_3 \ \equiv \ w_1 \wedge (w_2 \wedge w_3) \equiv \ w_1 \wedge w_2 \wedge w_3$ [conjunción]

  $(w_1 \vee w_2) \vee w_3 \ \equiv \ w_1 \vee (w_2 \vee w_3) \equiv \ w_1 \vee w_2 \vee w_3$ [disyunción]
- **Leyes distributivas:**

  $w_1 \wedge (w_2 \vee w_3) \ \equiv \ (w_1 \wedge w_2) \vee (w_1 \wedge w_3)$

  $w_1 \vee (w_2 \wedge w_3) \ \equiv \ (w_1 \vee w_2) \wedge (w_1 \vee w_3)$

Si w_1 y w_2 son equivalentes, entonces $(w_1 \Leftrightarrow w_2)$ es válido

Consecuencia lógica

⌘ **Consecuencia lógica:** Si una fbf w tiene valor *Verdadero* bajo todas las interpretaciones en las que todas las fbfs de Δ tienen valor *Verdadero*, entonces decimos que

- w es consecuencia lógica de Δ

$$\Delta \models w$$

(también "w se sigue de Δ")

Ejemplos: $\{P\} \models P$

$\{P, P \Rightarrow Q\} \models Q$

$F \models w$ (siendo w cualquier fbf)

Corrección y completitud

- **Corrección:**
  Si para todo conjunto de fbfs Δ y toda fbf w
  $$\Delta \vdash_R w \quad \text{implica} \quad \Delta \models w$$
  se dice entonces que el **conjunto de reglas *R*** es **correcto**.

- **Completitud:**
  Si para todo conjunto de fbfs Δ y toda fbf w
  $$\Delta \models w \quad \text{implica} \quad \Delta \vdash_R w$$
  se dice entonces que el **conjunto de reglas *R*** es **completo**.

- Para determinar si **w se sigue de** Δ ($\Delta \models w$)
  - Comprobación de modelos (uso de la definición): Construir **tablas de verdad** para todas las interpretaciones posibles (costoso: $O(2^n)$, n = número de átomos)
  - De manera alternativa, si tenemos un conjunto de reglas ***R*** **correcto y completo, podemos buscar una prueba** usando un **procedimiento de búsqueda completa:**
    - Las reglas de inferencia son los operadores que generan los sucesores del estado actual de búsqueda.
    - Normalmente se necesita transformar las expresiones a formas normales.

Metalingüística

No confundir:

- **Símbolos lingüísticos** que forman frases en lógica proposicional:
  V, F, P, Q, P1, P2…, $\vee$, $\wedge$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
- **Símbolos metalingüísticos**, usados para realizar afirmaciones sobre lógica proposicional:
  $\models$ (consecuencia lógica), $\vdash$ (inferencia)

No confundir:

- **Teoremas en lógica proposicional**, producidas por la aplicación de reglas de inferencia a conjuntos de fbfs.
- **Metateoremas:** Teoremas sobre lógica proposicional

Pregunta: ¿Es un teorema o un metateorema la siguiente afirmación (correcta)?

“Las reglas de equivalencia son reglas de inferencia correctas”

¿Se cumple $\Delta \models$ w ?

Establecer si todos los modelos para el conjunto de fbfs Δ son también modelos para una fbf w

- **Comprobación de modelos** (enumeración de modelos)
  - » Asignar todas las posibles combinaciones de *Verdadero* y *Falso* a los átomos en Δ (computacionalmente costoso: con n átomos hay 2^n combinaciones)
  - » Chequear si en toda asignación donde todas las fórmulas en Δ tienen valor *Verdadero*, w tiene también valor *Verdadero.*

- Solución **buscando una prueba :**

  Si podemos encontrar un conjunto de reglas de inferencia *R* correcto y completo, podemos determinar si w se sigue de Δ encontrando una prueba de w a partir de Δ

$$\Delta \vdash_R w$$

  - » Usar una estrategia de búsqueda completa para buscar en el espacio de conjuntos de fbfs.
  - » Las reglas de inferencia son los operadores que generan los sucesores del estado actual de búsqueda.

Metateoremas para la consecuencia lógica

- **<u>Metateorema:</u> Teorema de deducción**
  Si $\{w_1, w_2, ..., w_{n-1}, w_n\} \models w$, entonces $(w_1 \wedge w_2 \wedge ... \wedge w_{n-1} \wedge w_n) \Rightarrow w$ es **válido** y viceversa

- **<u>Metateorema:</u> Reducción al absurdo (contradicción)**
  w es consecuencia lógica del conjunto de fbfs Δ, w $\Delta \models w$ sii $\Delta \cup \{\neg w\}$ no tiene modelos; es decir, no hay ninguna interpretación que satisface simultáneamente Δ y $\neg w$)

  $\{\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_{n-1}}, \mathbf{w_n}\} \models \mathbf{w}$ puede demostrarse por

  - **Deducción**
    $(\mathbf{w_1} \wedge \mathbf{w_2} \wedge ... \wedge \mathbf{w_{n-1}} \wedge \mathbf{w_n}) \Rightarrow \mathbf{w}$ es una fbf válida
    (es decir, tiene el valor *Verdadero* bajo todas las interpretaciones)
  - **Reducción al absurdo (contradicción):**
    $(\mathbf{w_1} \wedge \mathbf{w_2} \wedge ... \wedge \mathbf{w_{n-1}} \wedge \mathbf{w_n} \wedge \neg \mathbf{w})$ es una fbf insatisfacible
    (es decir, no hay ninguna interpretación que haga su valor *Verdadero*)

Forma Normal Conjuntiva

- **Un literal** es un átomo (literal positivo), o un átomo precedido por el signo de negación $\neg$ (literal negativo).

  Ejemplos: Átomo, literal positivo: P. Literal negativo: $\neg$P

  Notación: Si λ es un literal negativo (eg. $\lambda=\neg P$), entonces se entiende que $\neg\lambda$ es el correspondiente literal positivo (P).

- **Cláusula**: Una cláusula K es una disyunción de literales
  $K = \bigvee_{j=1}^{k} \lambda_j$, donde $\lambda_1, \lambda_2, \ldots \lambda_k$ son m literales
  - **Cláusula unitaria**: Cláusula que contiene un sólo literal, $K=\lambda$
  - **Cláusula vacía** (que se denota con $\square$): No contiene ningún literal.
    - **La cláusula vacía es insatisfacible: $\square \equiv$ F**

- **Forma normal conjuntiva (FNC):** Una fbf α consistente en una conjunción (o conjunto) de cláusulas se dice que está en forma normal conjuntiva.
  CNF=conjunción de disyunciones de literales
  $\alpha = K_1 \wedge K_2 \wedge \ldots \wedge K_n = \wedge_{i=1}^{n} K_i = \{K_i\}_{i=1}^{n}$
  - El **FNC vacío** (que se denota por $\{\}$) no contiene ninguna cláusula.
    - **El FNC vacío es una tautología: { } $\equiv$ T**

Conversión a FNC

Metateorema: Hay un algoritmo que transforma cualquier fbf de cálculo proposicional en una FNC **equivalente:**

1. **Eliminación de las conectivas de implicación** usando

$$w_1 \Rightarrow w_2 \equiv \neg w_1 \vee w_2$$
$$w_1 \Leftrightarrow w_2 \equiv (w_1 \Rightarrow w_2) \wedge (w_2 \Rightarrow w_1)$$

2. **Reducir el ámbito de ¬**

   - Usando las leyes de De Morgan
   - Eliminando símbolos ¬ repetidos.

     Aplicar esta regla repetidamente hasta que todos los símbolos ¬ aparezcan sólo inmediatamente antes de un átomo

3. **Convertir a FNC** usando **leyes asociativas/distributivas.**

4. **Simplificar** la expresión usando **reglas de equivalencia**.

Ejemplo 1: Conversión a FNC

- **Considera la proposición**
  $((P \Leftrightarrow Q) \Rightarrow (R \Rightarrow S)) \wedge (Q \Rightarrow \neg (P \wedge R))$

- Convertir la proposición a una **FNC equivalente**

  1. Eliminación de los símbolos de implicación
     $(\neg ((\neg P \vee Q) \wedge (\neg Q \vee P)) \vee (\neg R \vee S)) \wedge (\neg Q \vee \neg (P \wedge R))$

  2. Reducción del ámbito de $\neg$
     $((\neg (\neg P \vee Q) \vee \neg (\neg Q \vee P)) \vee (\neg R \vee S)) \wedge (\neg Q \vee \neg P \vee \neg R)$

     $(((\neg\neg P \wedge \neg Q) \vee (\neg\neg Q \wedge \neg P))) \vee (\neg R \vee S)) \wedge (\neg Q \vee \neg P \vee \neg R)$
     $(((P \wedge \neg Q) \vee (Q \wedge \neg P) \vee (\neg R \vee S))) \wedge (\neg Q \vee \neg P \vee \neg R)$

Ejemplo 1: Conversión a FNC

3. Aplicar leyes asociativas / distributivas

$((P\wedge\neg Q)\vee(Q\wedge\neg P)\vee(\neg R\vee S))\wedge(\neg Q\vee\neg P\vee\neg R)$

[a continuación se genera una matriz equivalente a $((P\wedge\neg Q)\vee(Q\wedge\neg P)\vee(\neg R\vee S))$. En ella hay disyunciones implícitas entre filas, y conjunciones implícitas entre elementos de la misma fila]

$$\begin{array}{cc} P & \neg Q \\ Q & \neg P \\ \neg R & \\ S & \end{array}$$

(generar de esta matriz 8 conjunciones de disyunciones, donde cada disyunción tiene un elemento de cada fila)

$(P\vee Q\vee\neg R\vee S)\wedge(P\vee\neg P\vee\neg R\vee S)\wedge(\neg Q\vee Q\vee\neg R\vee S)\wedge\ (\neg Q\vee\neg P\vee\neg R\vee S)\wedge(\neg Q\vee\neg P\vee\neg R)$

4. Simplificación usando reglas de equivalencia

$(P\vee Q\vee\neg R\vee S)\wedge(\neg Q\vee\neg P\vee\neg R\vee S)\wedge(\neg Q\vee\neg P\vee\neg R)$

Ejemplo 2: Conversión a FNC

- **Considera la proposición**
  $((P \Leftrightarrow Q) \Rightarrow (\neg R \wedge S)) \wedge (Q \Rightarrow \neg (P \wedge R))$

- Convertir la proposición a una **FNC equivalente**

  1. Eliminación de los símbolos de implicación

$(\neg((\neg P \vee Q) \wedge (\neg Q \vee P)) \vee (\neg R \wedge S)) \wedge (\neg Q \vee \neg (P \wedge R))$

  2. Reducción del ámbito del $\neg$

$((\neg(\neg P \vee Q) \vee \neg(\neg Q \vee P)) \vee (\neg R \wedge S)) \wedge (\neg Q \vee \neg P \vee \neg R)$

$((\neg\neg P \wedge \neg Q) \vee (\neg\neg Q \wedge \neg P)) \vee (\neg R \wedge S)) \wedge (\neg Q \vee \neg P \vee \neg R)$

$(((P \wedge \neg Q) \vee (Q \wedge \neg P)) \vee (\neg R \wedge S)) \wedge (\neg Q \vee \neg P \vee \neg R)$

Ejemplo 2: Conversión a FNC

3. Aplicar leyes asociativas / distributivas

$((P\wedge\neg Q)\vee(Q\wedge\neg P)\vee(\neg R\wedge S))\wedge(\neg Q\vee\neg P\vee\neg R)$

P	$\neg Q$	(generar de esta matriz 8 conjunciones de
Q	$\neg P$	disyunciones, donde cada disyunción tiene
$\neg R$	S	un elemento de cada fila)

$(P\vee Q\vee\neg R)\wedge(P\vee Q\vee S)\wedge(P\vee\neg P\vee\neg R)\wedge(P\vee\neg P\vee S)\wedge$
$(\neg Q\vee Q\vee\neg R)\wedge(\neg Q\vee Q\vee S)\wedge(\neg Q\vee\neg P\vee\neg R)\wedge(\neg Q\vee\neg P\vee S)\wedge$
$(\neg Q\vee\neg P\vee\neg R)$

4. Simplificar la expresión

$(P\vee Q\vee\neg R)\wedge(P\vee Q\vee S)\wedge(\neg Q\vee\neg P\vee\neg R)\wedge(\neg Q\vee\neg P\vee S)$

Metateoremas para formas normales

- **<u>Metateorema</u> :** Sean $\lambda_1, \lambda_2, \ldots \lambda_n$ n literales diferentes.
  Las siguientes afirmaciones son equivalentes:

  » $\vee_{i=1}^{n} \lambda_i$ es válido
  » $\wedge_{i=1}^{n} \lambda_j$ es insatisfacible
  » Algún par (λ_i, λ_j) $1 \leq i, j \leq n$ es un par de literales complementarios o “parejas” $(\lambda_i = \neg\lambda_j)$

Demostración

» Es obvio que si $\lambda_i = \neg\lambda_j$, entonces $\vee_{i=1}^{n} \lambda_i$ es válido y $\wedge_{i=1}^{n} \lambda_i$ es insatisfacible.

» Si no hay ningún par de parejas en $\lambda_1, \lambda_2, \ldots \lambda_n$ entonces

  » En la interpretación donde todos $\{\lambda_i\}_{i=1}^{n}$ tienen valor *Verdadero*, $\wedge_{i=1}^{n} \lambda_i$ tiene valor *Verdadero*.
  » En la interpretación donde todos $\{\lambda_i\}_{i=1}^{n}$ tienen valor *Falso*, $\vee_{i=1}^{n} \lambda_i$ tiene valor *Falso*.

Metateoremas para formas normales

- **<u>Metateorema:</u>** Una fórmula en FNC es una tautología sii cada una de sus cláusulas es una tautología.

Demostración

» Sea α una fbf en FNC

» Sean K_i, $i=1,2,...,n$ las cláusulas en α

$$\alpha = K_1 \wedge K_2 \wedge ... \wedge K_n = \wedge_{i=1}^{n} K_i$$

» Si cada K_i es una tautología entonces, para cualquier interpretación,

K_i $i=1,2,...,n$ tienen valor *Verdadero*, y entonces $\wedge_{i=1}^{n} K_i$ tiene valor *Verdadero*.

» Si algún K_i no es una tautología, entonces existe al menos una interpretación donde K_i tiene el valor *Falso*. En esa interpretación $\wedge_{i=1}^{n} K_i$ tiene el valor *Falso*.

Resolución sobre cláusulas

La resolución es una regla de inferencia

- Sea λ un átomo y Σ_1, Σ_2 unas cláusulas (disyunciones de literales) que no contienen ni λ ni $\neg\lambda$:
  $\Sigma_1 \vee \Sigma_2$ puede ser inferido de $\lambda \vee \Sigma_1$ y $\neg\lambda \vee \Sigma_2$

  » **λ es el átomo sobre el que se resuelve**.

  » **$\Sigma_1 \vee \Sigma_2$ es una cláusula**, que se conoce como el **resolvente** de las dos cláusulas $\lambda \vee \Sigma_1$ y $\neg\lambda \vee \Sigma_2$

$$\boxed{\mathbf{res}_\lambda(\lambda \vee \Sigma_1, \neg\lambda \vee \Sigma_2) = \Sigma_1 \vee \Sigma_2}$$

Casos particulares de resolución:

» **Encadenamiento**: De $R \Rightarrow P$ y $P \Rightarrow Q$ se infiere $R \Rightarrow Q$
Resolviendo $\neg R \vee P$ y $\neg P \vee Q$ se obtiene $\neg R \vee Q$

» ***Modus ponens*:** De R y $R \Rightarrow P$ se infiere P
Resolviendo R y $\neg R \vee P$ se obtiene P

» Resolviendo $P \vee Q \vee R \vee S$ y $\neg P \vee Q \vee W$ se obtiene $Q \vee R \vee S \vee W$
Nota: Q no aparece dos veces (eliminación de repeticiones).

» Las fbfs $P \vee Q \vee \neg R$ y $P \vee W \vee \neg Q \vee R$ dan

» Resolviendo sobre Q: $P \vee \neg R \vee R \vee W \quad \equiv \quad T$

» Resolviendo sobre R: $P \vee Q \vee \neg Q \vee W \quad \equiv \quad T$

La resolución es correcta

Corrección: $\Delta \vdash_R w$ implica $\Delta \models w$

***R* = {*resolución sobre cláusulas*};** $\Delta = \{\lambda \vee \Sigma_1, \neg\lambda \vee \Sigma_2\}$; $w = \Sigma_1 \vee \Sigma_2$

$\Delta \models w$ si la fbf **w tiene valor *Verdadero*** bajo **todas las interpretaciones** en las que **todas las fbfs de Δ** tienen valor ***Verdadero***

λ	Σ_1	Σ_2	$\lambda \vee \Sigma_1$	$\neg\lambda \vee \Sigma_2$	$\Sigma_1 \vee \Sigma_2$
Verdadero	*Verdadero*	*Verdadero*	*Verdadero*	*Verdadero*	*Verdadero*
Verdadero	*Verdadero*	*Falso*	*Verdadero*	*Falso*	*Verdadero*
Verdadero	*Falso*	*Verdadero*	*Verdadero*	*Verdadero*	*Verdadero*
Verdadero	*Falso*	*Falso*	*Verdadero*	*Falso*	*Falso*
Falso	*Verdadero*	*Verdadero*	*Verdadero*	*Verdadero*	*Verdadero*
Falso	*Verdadero*	*Falso*	*Verdadero*	*Verdadero*	*Verdadero*
Falso	*Falso*	*Verdadero*	*Falso*	*Verdadero*	*Verdadero*

La resolución no es completa

¿Completitud?

¿$\Delta \models w$ implica $\Delta \vdash_R w$?

***R* = {*resolución sobre cláusulas*};**
$\Delta = \{\lambda \vee \Sigma_1 , \neg\lambda \vee \Sigma_2\}$; $w = \Sigma_1 \vee \Sigma_2$

- **La resolución no es completa:**
  No todas las consecuencias lógicas pueden ser generadas sólo por resolución.

  Ejemplo: La consecuencia lógica $\{P\} \models P \vee Q$ no puede deducirse del conjunto de cláusulas $\{P\}$

  Pero, mediante refutación por resolución:
  $\Delta = \{P, \neg(P \vee Q)\}$ puede demostrarse ser insatisfacible:

  - Usando equivalencia (ley de De Morgan) $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
  $\{P, \neg(P \vee Q)\} \equiv \{P, \neg P, \neg Q\}$

  - Usando resolución $\{P, \neg P\} \vdash_R \square$

Refutación por resolución

Dado un conjunto de fbfs Δ, y una fbf w

1. Convertir todas las fbfs de Δ en el conjunto de cláusulas equivalente.
2. Convertir $\neg w$ en un conjunto de cláusulas equivalente.
3. Combinar las cláusulas que resultan de los pasos 1 y 2 en un único conjunto de cláusulas α.
4. Aplicar todas las resoluciones posibles en el conjunto α, añadiendo las cláusulas resultantes a α.
5. Iterar el paso (4) hasta que no se puedan añadir más cláusulas nuevas, o hasta que se genere la cláusula vacía.

- **La refutación por resolución es completa:**
  Si $\Delta \models w$, existe una derivación por resolución de la cláusula vacía a partir de $\Delta \cup \{\neg w\}$
  La resolución en lógica proposicional es completa si se utiliza refutación como método de demostración.

- **El cálculo proposicional es decidible mediante refutación por resolución:**
  Sea Δ un conjunto de fbfs y sea w una fbf tal que $\Delta \not\models w$.
  Entonces el procedimiento de refutación por resolución termina sin producir la cláusula vacía.

Cláusulas y fbfs en FNC como conjuntos

- Una cláusula K es el conjunto $\{\lambda_j\}_{j=1}^{k}$ (también $\vee_{j=1}^{k} \lambda_j$) donde $\lambda_1, \lambda_2, ... \lambda_k$ son m literales diferentes
  - » Cláusula unitaria: Una cláusula que contiene un sólo literal $\{\lambda\}$
  - » Cláusula vacía (que se denota con □): Una cláusula sin literales
    - » La cláusula vacía es insatisfacible.
- Sea α una fbf en FNC: $\alpha = K_1 \wedge K_2 \wedge ... \wedge K_n = \wedge_{i=1}^{n} K_i$
  α puede ser representada por el conjunto de cláusulas $\{K_i\}_{i=1}^{n}$
  - » Una fbf vacía en FNC (denotada como {}) es una tautología.
  - » Si K es una tautología, y $K \in \alpha$, entonces α es satisfacible sii $\alpha - \{K\}$ es satisfacible.

  Importante:
  - » El orden y las repeticiones se pierden utilizando la representación como conjuntos.
    Para valores de verdad esto es relevante ya que $\vee, \wedge$ son conmutativos e idempotentes.
  - » Si λ es un literal negativo (es decir $\lambda = \neg P$, P es un átomo), entonces se entiende que $\neg\lambda$ es el literal equivalente P.
  - » Si queremos demostrar la satisfacibilidad de α, se pueden eliminar sistemáticamente las tautologías ya que son irrelevantes en determinar la satisfacibilidad de α.

Definiciones

- El resolvente de dos cláusulas K_1, K_2 bajo el literal λ es
  $res_\lambda(K_1, K_2) = \{K_1 - \{\lambda\}\} \cup \{K_2 - \{\neg\lambda\}\}$ $\lambda \in K_1 ; \neg\lambda \in K_2$

- Sea λ un literal, K una cláusula,
  - » K es λ-neutro si $\lambda \notin K, \neg\lambda \notin K$.
  - » K es λ-positivo si $\lambda \in K$
  - » K es λ-negativo si $\neg\lambda \in K$

  Una cláusula K es una tautología sii es tanto λ-positiva como λ-negativa para algún literal λ.

» **IMPORTANTE:** De aquí en adelante **se eliminan las tautologías de todos los conjuntos de cláusulas.**

Definiciones

- Sea λ un literal, α un conjunto de cláusulas,
  - » $\alpha = \alpha_\lambda^0 \cup \alpha_\lambda^+ \cup \alpha_\lambda^-$

$\alpha_\lambda^0 = \alpha^0(\lambda) = \{K \in \alpha \ / \lambda \notin K, \neg\lambda \notin K\}$ (los Ks son λ-neutros)

$\alpha_\lambda^+ = \alpha^+(\lambda) = \{K \in \alpha \ / \ \lambda \in K\}$ (los Ks son λ-positivos)

$\alpha_\lambda^- = \alpha^-(\lambda) = \{K \in \alpha \ / \neg\lambda \in K\}$ (los Ks son λ-negativos)

Si se eliminan las tautologías, entonces $\alpha_\lambda^+ \cap \alpha_\lambda^-$ es vacío.

» $\mathrm{POS}_\lambda(\alpha) = \mathrm{POS}(\alpha;\lambda) = \alpha_\lambda^0 \cup \{\{K-\{\lambda\}\} \ / \ K \in \alpha_\lambda^+\}$

» $\mathrm{NEG}_\lambda(\alpha) = \mathrm{NEG}(\alpha;\lambda) = \alpha_\lambda^0 \cup \{\{K-\{\neg\lambda\}\} / \ K \in \alpha_\lambda^-\}$

» $\mathrm{RES}_\lambda(\alpha) = \mathrm{RES}(\alpha;\lambda) = \alpha_\lambda^0 \cup \{\mathrm{res}_\lambda(K_1, K_2) / K_1 \in \alpha_\lambda^+, K_2 \in \alpha_\lambda^-\}$

Reglas de eliminación que pueden ser usadas en estrategias de resolución

- **Eliminar literales repetidos en cláusulas.**
- **Eliminar cláusulas repetidas.**
- **Eliminar cláusulas que son tautologías.**
- **Eliminar cláusulas subsumidas.**
  - Si $K_1 \subset K_2$, entonces decimos que K_2 está **subsumido** por K_1 (K_1 subsume a K_2)
  - Si $K_1 \subset K_2$, entonces $K_1 \models K_2$
  - Si ambos $K_1, K_2 \in \alpha$ y K_2 es subsumido por K_1 ($K_1 \subset K_2$) entonces α es satisfacible sii $\alpha - \{K_2\}$ es satisfacible.

Ejemplo:

$K_1 = P \vee Q \vee R$ $K_2 = P \vee Q \vee R \vee S$

$K_1 \subset K_2$ (K_1 subsume a K_2)

Si ambas cláusulas están presentes en α, entonces podemos eliminar K_2 de α

Reglas de Davis-Putnam

- **<u>Metateorema R1</u> (regla de desdoblamiento)**: Sea λ un literal.

  El conjunto de fbfs α en FNC es satisfacible sii al menos un miembro del par $POS_\lambda(\alpha)$, $NEG_\lambda(\alpha)$ es satisfacible.

<u>Demostración</u>:

» Si α es satisfacible, entonces existe una interpretación bajo la cual α tiene el valor *Verdadero*.

  » Si bajo esta interpretación λ tiene el valor *Verdadero* entonces $NEG_\lambda(\alpha)=\alpha_\lambda^0\cup\{\{K-\{\neg\lambda\}\}/K\in\alpha_\lambda^-\}$ tiene el valor *Verdadero*

  » Si bajo esta interpretación λ tiene el valor *Falso* entonces $POS_\lambda(\alpha)=\alpha_\lambda^0\cup\{\{K-\{\lambda\}\}/K\in\alpha_\lambda^+\}$ tiene el valor *Verdadero*

» Si $POS_\lambda(\alpha)$ es satisfacible, se construye un modelo para α con el modelo de $POS_\lambda(\alpha)$ + la asignación para λ de *Falso*.

» Si $NEG_\lambda(\alpha)$ es satisfacible, se construye un modelo para α con el modelo de $NEG_\lambda(\alpha)$ + la asignación para λ de *Verdadero*.

Reglas de Davis-Putnam

- Corolario I (**regla del literal puro**):

Si el literal λ aparece o bien siempre negado, o bien siempre sin negar, entonces

α es satisfacible sii $\alpha_{\lambda}{}^{0}$ es satisfacible

- Corolario II (**regla de la cláusula unitaria**):

Si la cláusula unitaria $\{\lambda\} \in \alpha$ entonces α es satisfacible sii $NEG_{\lambda}(\alpha)$ es satisfacible.

Ejemplo:
Uso de las reglas Davis-Putnam

¿Es α satisfacible?

$\alpha = \{\neg P \vee \neg Q \vee \neg R, \neg P \vee S, P \vee S, Q \vee R, \neg Q \vee S, \neg Q \vee P, P, Q, \neg R, \neg P \vee \neg Q \vee R\}$

⌘ Uso de la **regla del literal puro**: Comprobar la satisfacibilidad sólo para aquellas cláusulas que no contienen los literales puros.

S es un literal puro

$\alpha_1 = \{\neg P \vee \neg Q \vee \neg R, Q \vee R, \neg Q \vee P, P, Q, \neg R, \neg P \vee \neg Q \vee R\}$

⌘ Uso de la **regla de la cláusula unitaria:** Comprobar la satisfacibilidad de $NEG_\lambda(\alpha)$ para aquellos literales λ que aparecen en cláusulas unitarias.

$\alpha_2 = NEG_Q(\alpha_1) = \{\neg P \vee \neg R, P, \neg R, \neg P \vee R\}$

$\alpha_3 = NEG_{\neg R}(\alpha_2) = POS_R(\alpha_2) = \{P, \neg P\}$

$\alpha_4 = NEG_P(\alpha_2) = \{\square\}$

α es insatisfacible

Cierre de un conjunto de cláusulas bajo resolución

El **cierre** de un **conjunto de cláusulas α bajo resolución**, **RC(α),** es el conjunto de **todas las cláusulas** que pueden ser **deducidas aplicando repetidamente resolución**.

<u>Propiedades:</u>

» **RC(α)** es **finito** (α contiene un conjunto finito de átomos)
» **RC(α) es cerrado bajo resolución**.

De los metateoremas de resolución:

» **RC(α)** contiene la cláusula vacía sii el conjunto de cláusulas α es insatisfacible.
» **RC(α)** no contiene la cláusula vacía sii el conjunto de cláusulas α es satisfacible.

Resolución por saturación de nivel

⌘ **Resolución por saturación de nivel:**

- $\alpha_0 = \alpha$
- $\alpha_i = \{$todos los posibles resolventes entre α_{i-1} y $\alpha_1 \cup \alpha_2 \cup ... \cup \alpha_{i-1}\}$, $i=1,2,3,...$

  hasta que no se puedan añadir más resolventes o hasta que se produce la cláusula vacía.

⌘ **Ejemplo:**

$\alpha = \{P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$

Nota: tanto α como las cláusulas en α están definidos como conjuntos (i.e. el orden no es importante; las cláusulas repetidas en α pueden ser listadas sólo una vez; los literales repetidos en una cláusula pueden ser listados como un sólo literal).

Las cláusulas que son tautologías pueden ser eliminadas de la BC.

- $\alpha_0 = \alpha = \{P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$
- $\alpha_1 = \{Q \vee Q, P \vee P, Q \vee \neg Q, P \vee \neg P, Q \vee \neg Q,$
  $P \vee \neg P, \neg P \vee \neg P, \neg Q \vee \neg Q\} \equiv \{Q, P, \neg P, \neg Q\}$

  $\alpha_0 \cup \alpha_1 = \{P, Q, \neg P, \neg Q, P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$

- $\alpha_2 = \{\square, P, Q, \neg P, \neg Q, P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$

Una estrategia directa de comprobar SAT

¿Es un conjunto de cláusulas α SAT?

Sean $\lambda_1, \lambda_2, \ldots \lambda_{k-1}, \lambda_k$ átomos diferentes que forman las cláusulas de α.

Sean $\alpha_0 = \alpha$,

$\alpha_j = \mathrm{RES}(\alpha_{j-1}; \lambda_j)$, (se eliminan las tautologías de α_j), $j=1,2\ldots,k$

hasta que no se puedan aplicar más resoluciones o hasta que se genere la cláusula vacía.

<u>Ejemplo</u>:

$\alpha = \{P \vee Q, \neg P \vee Q, P \vee \neg Q\}$

La lista de los diferentes átomos de α es P, Q.

- $\alpha_0 = \alpha = \{P \vee Q, \neg P \vee Q, P \vee \neg Q\}$
  - $\alpha^0(P) = \{\}$
  - $\alpha^+(P) = \{P \vee Q, P \vee \neg Q\}$
  - $\alpha^-(P) = \{\neg P \vee Q\}$
- $\alpha_1 = \mathrm{RES}(\alpha_0; P) = \{Q, Q \vee \neg Q\} \equiv \{Q\}$
  - $\alpha_1{}^0(Q) = \{\}$
  - $\alpha_1{}^+(Q) = \{Q\}$
  - $\alpha_1{}^-(Q) = \{\}$
- $\alpha_2 = \mathrm{RES}(\alpha_1; Q) = \{\}$ (conjunto vacío de cláusulas)

Luego α es SAT

Una estrategia directa de comprobar SAT

<u>Ejemplo</u>:

$\alpha = \{P \lor Q, \neg P \lor Q\ , P \lor \neg Q, \neg P \lor \neg Q\}$

La lista de los diferentes átomos de α es P, Q.

- $\alpha_0 = \alpha = \{P \lor Q, \neg P \lor Q\ , P \lor \neg Q, \neg P \lor \neg Q\}$
  - $\alpha^0(P) = \{\}$
  - $\alpha^+(P) = \{P \lor Q, P \lor \neg Q\}$
  - $\alpha^-(P) = \{\neg P \lor Q\ \neg P \lor \neg Q\}$
- $\alpha_1 = RES(\alpha_0; P) = \{Q, Q \lor \neg Q, \neg Q\} \equiv \{Q, \neg Q\}$
  - $\alpha_1{}^0(Q) = \{\}$
  - $\alpha_1{}^+(Q) = \{Q\}$
  - $\alpha_1{}^-(Q) = \{\neg Q\}$
- $\alpha_2 = RES(\alpha_1; Q) = \{\square\}$ (conjunto con la cláusula vacía)

Luego α es INSAT

Ejemplo

¿Se cumple $\Delta \models w$?

$$\Delta = \{P \Leftrightarrow (Q \lor R), \neg P\}$$

$$w = \neg Q$$

- Conversión a FNC

  $\Delta \equiv \{\neg P \lor Q \lor R, \neg Q \lor P, \neg R \lor P, \neg P\}$

  $\neg w \equiv Q$
- Construcción de α

  $\alpha = \{\neg P \lor Q \lor R, \neg Q \lor P, \neg R \lor P, \neg P, Q\}$
- Uso de metateoremas R2 y R4, eliminando tautologías

  La lista de los diferentes átomos de α es P, Q, R.
  - $\alpha_0 = \alpha = \{\neg P \lor Q \lor R, \neg Q \lor P, \neg R \lor P, \neg P, Q\}$
    - $\alpha^0(P) = \{Q\}$
    - $\alpha^+(P) = \{\neg Q \lor P, \neg R \lor P\}$
    - $\alpha^-(P) = \{\neg P \lor Q \lor R, \neg P\}$
  - $\alpha_1 = RES(\alpha_0; P) = \{Q, \neg Q \lor Q \lor R, \neg Q, \neg R \lor Q \lor R, \neg R\} \equiv \{Q, \neg Q, \neg R\}$
    - $\alpha^0(Q) = \{\neg R\}$
    - $\alpha^+(Q) = \{Q\}$
    - $\alpha^-(Q) = \{\neg Q\}$
  - $\alpha_2 = RES(\alpha_1; Q) = \{\neg R, \square\}$

α es insatisfacible

Resolución unitaria

Forma de resolución restrictiva donde en cada resolución se utiliza una cláusula unitaria (cláusula con un sólo literal)

- **Eficiente**
- **No completa** en general
- Completa para bases de conocimiento de Horn

Ejemplo: $\alpha = \{P\vee Q, \neg P\vee R, \neg Q\vee R, \neg R\}$

```
(1) P∨Q     (2) ¬P∨R     (3) ¬Q∨R     (4) ¬R
-----------
(5) {¬P} [(2)+(4)]
(6) {¬Q} [(3)+(4)]
-----------
(7) {Q} [(1)+(5)]
(8) {P} [(1)+(6)]
-----------
(10) {R} [(3)+(7)]
(11) {□} [(6)+(7)]
(12) {R} [(2)+(8)]
(13) {□} [(5)+(8)]
```

⌘ **Preferencia por resolución unitaria (siempre que sea posible):** preferir resoluciones con cláusulas unitarias. **¿Por qué?:** Como estamos intentando generar la cláusula vacía, preferimos inferencias que generan cláusulas más pequeñas.

- Introducida por primera vez en 1964.
- La mejora computacional no es suficiente para resolver problemas de tamaño medio.

Resolución por conjunto soporte

- El **conjunto soporte** de un conjunto de cláusulas α es un subconjunto Γ tal que el conjunto $\alpha-\Gamma$ es satisfacible.

- **Resolución por conjunto soporte**
  Es una forma restrictiva de resolución en la cual cada paso de resolución de realizarse con por lo menos una cláusula del conjunto soporte.
  La cláusula resolvente es entonces incluida en Γ.

  - **Completo si $\alpha-\Gamma$ es satisfacible**. No es completo si se elige mal el conjunto soporte inicial.
  - En problemas de (¿ $\Delta \models w$?, $\alpha = \{\Delta \wedge \neg w\}$), una elección común es inicializar Γ con el resultado de negar la query inicial ($\Gamma = \{\neg w\}$)
    - Esta elección es correcta si la base de conocimiento original Δ es consistente (satisfacible). Nótese que si la BC original no es consistente, cualquier afirmación puede ser derivada de ella.
    - Es una refutación dirigida por objetivos (inteligible!)

<u>Ejemplo</u>: $\Delta = \{P \vee Q, \neg P \vee R, \neg Q \vee R\}$; $w = R$.

$\alpha = \{P \vee Q, \neg P \vee R, \neg Q \vee R, \neg R\}$

(0) $\Gamma_0 = \{\neg R\}$;

(1) $\Gamma_1 = \{\neg R, \neg P\}$;

(2) $\Gamma_2 = \{\neg R, \neg P, \neg Q\}$;

(3) $\Gamma_3 = \{\neg R, \neg P, \neg Q, Q\}$;

(4) $\Gamma_4 = \{\neg R, \neg P, \neg Q, Q, \square\}$;

Resolución de entrada y resolución lineal

- **Resolución de entrada (Input resolution)**

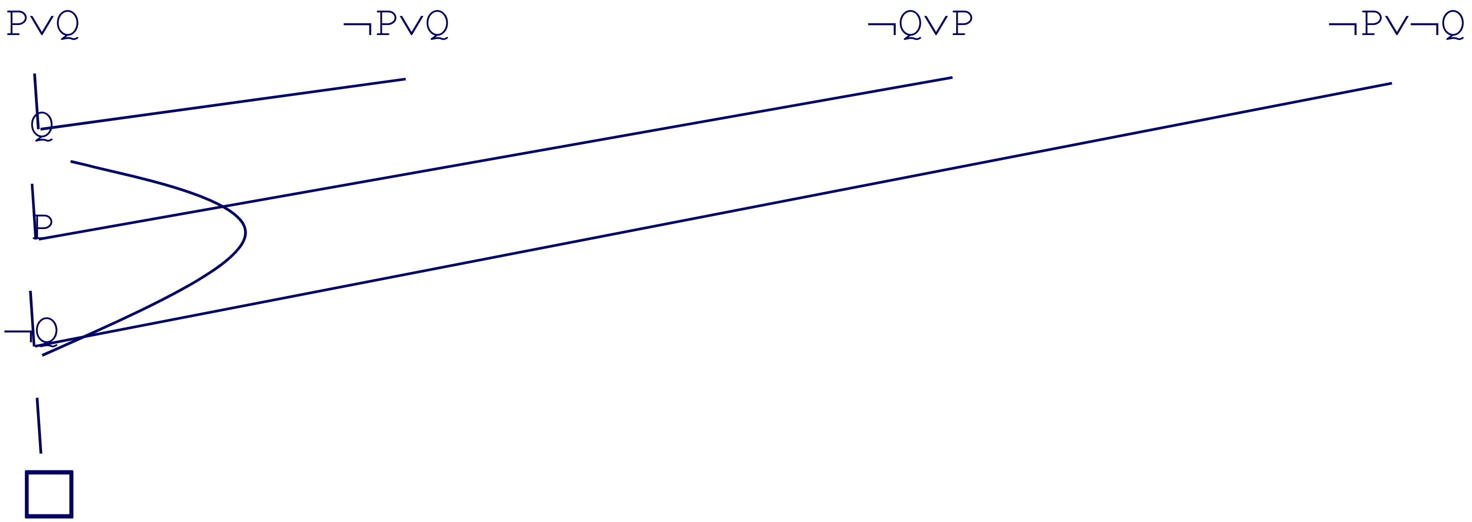
  De las dos cláusulas usadas en cada paso de resolución, una siempre pertenece al conjunto inicial de cláusulas.
  - En general **no es completa**
  - Completa para bases de conocimiento en forma de Horn.

- **Resolución lineal (ancestry-filtered resolution)**

  Generalización de la resolución de entrada. De las dos cláusulas usadas en cada paso de resolución, una siempre pertenece al conjunto inicial de cláusulas o es antecesora de la cláusula actual del árbol de derivación
  - **Completa.**

<u>Ejemplo</u>: $\alpha = \{P\vee Q,\ \neg P\vee Q,\ \neg Q\vee P,\ \neg P\vee\neg Q\}$

La resolución de entrada falla



Cláusulas de Horn

Una **cláusula de Horn** es un conjunto no vacío de literales en el que **no hay más de un literal positivo**.

- Una **cláusula de Horn** es o bien
  - **Una restricción:** una cláusula **sin ningún literal positivo** (es decir, la negación de una cláusula objetivo conjuntiva)

    Ej. $\neg P \vee \neg Q \equiv \neg(P \wedge Q) \equiv (P \wedge Q) \Rightarrow F$

  - **Un "hecho"**: una cláusula unitaria formada por un solo literal positivo

    Ej. P, Q (hechos)

  - **Una "regla":** una cláusula con un solo literal positivo y con al menos un literal negativo.

    Una regla puede ser escrita como una **implicación**, donde la premisa es una conjunción de las negaciones de los literales negativos, y el consecuente es el literal positivo.

    Ej. $\neg P \vee \neg Q \vee \neg R \vee S \equiv (P \wedge Q \wedge R) \Rightarrow S$ (regla)

- La inferencia sobre un conjunto α de cláusulas de Horn puede ser hecha por
  - » **Encadenamiento hacia delante** (razonamiento dirigido por los datos)
  - » **Encadenamiento hacia atrás** (razonamiento dirigido por el objetivo)

» El problema **SAT** sobre un conjunto de **cláusulas de Horn es lineal** en el número de cláusulas del conjunto.

Encadenamiento hacia delante (EHD)

⌘ **Considérese un conjunto de cláusulas de Horn** α_0

1. $\alpha = \alpha_0$
2. Aplicar aquellas reglas cuyas premisas están en α, e incorporar las conclusiones a α.
3. Repetir (2) hasta que no se puedan aplicar más reglas.

⌘ Encadenamiento hacia delante (EHD) sobre conjuntos de cláusulas de Horn

- **EHD es correcto:** Cada inferencia es una aplicación del modus ponens, que es correcto.
- **EHD es completo:** EHD deriva todas las sentencias atómicas que son consecuencia lógica de α_0

Demostración:

- Cuando EHD termina no se pueden deducir más sentencias atómicas por aplicación de EHD
- Interpretación donde se asigna el valor *Verdadero* a todos los átomos deducidos por EHD de α_0 , y el valor *Falso* a todos los átomos que no se deducen por EHD de α_0 .
  Por contradicción: Asumamos que la regla $(P_1 \wedge P_2 \wedge ... \wedge P_n) \Rightarrow S$ de α_0 tiene valor *Falso*.
  $(P_1 \wedge P_2 \wedge ... \wedge P_n)$ debe entonces tener valor *Verdadero* y S debe tener valor *Falso.*
  Entonces $P_1, P_2, ..., P_n$ tienen valor *Verdadero* i.e. han sido deducidos por EHD, y la regla $(P_1 \wedge P_2 \wedge ... \wedge P_n) \Rightarrow S$ debería haber sido aplicada.
- Asumamos que existe un átomo $P / \alpha_0 \models P$ entonces P tiene el valor *Verdadero* en todos los modelos de α_0, y P debe ser uno de los átomos deducidos por EHD de α_0.

Encadenamiento hacia atrás (EHA)

⌘ **Deducir el objetivo** P **a partir de un conjunto** α **de cláusulas de Horn**

1. Si $P \in \alpha$ entonces objetivo encontrado.

2. Demostrar por encadenamiento hacia atrás de todos los átomos en la premisa de alguna regla en α cuya conclusión es P.

⌘ Evitar bucles:

Chequear si el nuevo subobjetivo ya está en la pila de objetivos.

⌘ Chequear si ya se ha demostrado que el nuevo subobjetivo:

- puede ser deducido a partir de α.
- no puede ser deducido a partir de α.

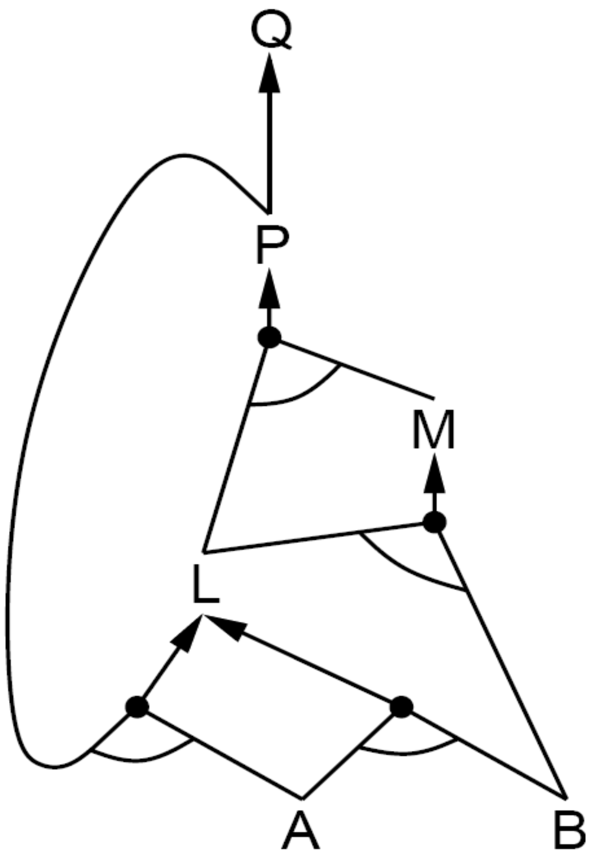
⌘ Encadenamiento hacia atrás vs Encadenamiento hacia adelante

- **EHD está** **dirigido por los datos**, apropiado para adquisición de conocimiento sin un objetivo específico en mente.
- **EHA está** **dirigido por el objetivo**, apropiado para resolver problemas específicos. La complejidad del EHA puede ser mucho menor que lineal (en tamaño de la BC).

Un agente debería usar EHD para generar hechos que probablemente serán relevantes para preguntas gestionadas por EHA

EHD/EHA: diagrama AND-OR

- Ejemplo (de Russel & Norvig)
  Considérese un agente cuya base de conocimiento es
  Hechos: `A,B`
  Reglas: $P \Rightarrow Q$, $L \wedge M \Rightarrow P$, $B \wedge L \Rightarrow M$, $A \wedge P \Rightarrow L$, $A \wedge B \Rightarrow L$



Consecuencia lógica por EHD/EHA

Considérese

Δ ={ P,N, P$\Rightarrow$Q, Q$\Rightarrow$R, R$\wedge$S$\Rightarrow$M, N$\Rightarrow$S}, w = M$\wedge$R

¿Se cumple $\Delta \models$ w ?

- Construir $\alpha = \Delta \cup \neg$w
- Nótese que $\neg$w es una cláusula de Horn

  $\neg$(M$\wedge$R)$\equiv$ $\neg$M$\vee\neg$R $\equiv$ M$\wedge$R$\Rightarrow$F
- Mostrar que α is insatisfacible por o bien:
  - Encadenamiento hacia delante:

    Hechos: {P,N}

    {P,N,Q,S} [por las reglas P$\Rightarrow$Q,N$\Rightarrow$S]

    {P,N,Q,S,R} [por la regla Q$\Rightarrow$R]

    {P,N,Q,S,R,M} [por la regla R$\wedge$S$\Rightarrow$M]

    {P,N,Q,S,R,M,F} [por la regla M$\wedge$R$\Rightarrow$F]
  - Encadenamiento hacia atrás:

    F puede ser inferido de M[1],R[2]

    (1) M puede ser inferido de R[1.1]=[2],S[1.2]

    (1.2) S puede ser inferido de N[1.2.1]

    (1.2.1) N está en la base de conocimiento.

    (2) R puede ser inferido de Q [2.1]

    (2.1) Q puede ser inferido de P [2.1.1]

    (2.1.1) P está en la base de conocimiento.

Comprobación de modelos para $\Delta \models w$

Establecer si todos los modelos para el conjunto de fbfs Δ son también modelos para una fbf `w`

- **Comprobación de modelos (enumeración de modelos):**
  - » Asignar todas las posibles combinaciones de *Verdadero* y *Falso* a los átomos en las fórmulas.
  - » Chequear si cada vez que todas las fórmulas en Δ tienen valor *Verdadero*, `w` tiene también valor *Verdadero*.
- Esta estrategia es
  - » Correcta: Aplicación directa de la definición de consecuencia lógica.
  - » Completa: Basada en una búsqueda exhaustiva en el espacio de interpretaciones (que es finito)
  - » Complejidad temporal: $O(2^n$; n = número de átomos)
  - » Complejidad espacial O(n)

Algoritmos:

- » **DPLL** (Davis, Putnam, Logemann, Loveland).
- » **GSAT.**
- » **WALKSAT** (variación estocástica, con camino aleatorio de GSAT)

El algoritmo DPLL

- **Davis, Putnam, Logemann, Loveland (1962)**
  - ¿Es un conjunto de cláusulas α satisfacible?
  - Recursivo, enumeración primero-en-profundidad de los posibles modelos.
  - **Terminación temprana**:
    - Una cláusula tiene valor *Verdadero* si cualquiera de sus literales tiene valor *Verdadero.*
    - Un conjunto de cláusulas tiene valor *Falso* si cualquiera de sus cláusulas tiene valor *Falso.*

  - Uso de las **reglas Davis-Putnam**
    - **Regla del literal puro**:
      - Si α_λ^- es vacío y α_λ^+ no, un modelo para α debe asignar a λ el valor *Verdadero.*
      - Si α_λ^+ es vacío y α_λ^- no, un modelo para α debe asignar a λ el valor *Falso*.
      -
    - **Regla de la cláusula unitaria**:
      - Si la cláusula unitaria $\{\lambda\} \in \alpha$ entonces un modelo para α debe asignar a λ el valor *Verdadero.*
      - Si en cualquier cláusula de α todos los literales excepto λ tienen asignado el valor *Falso*, entonces un modelo para α debe asignar a λ el valor *Cierto.*

Algoritmos de búsqueda local para SAT

¿Es el conjunto de cláusulas α satisfacible?

⌘ GSAT

1. Seleccionar un conjunto aleatorio de valores de verdad para todos los átomos en α.
2. Usar búsqueda local avariciosa para maximizar el número de cláusulas que tienen valor *Cierto* modificando en cada iteración el valor de un sólo átomo.
   - No es óptima (puede quedar atrapada en máximos locales).

Ejemplo: $\alpha = \{\neg P \vee \neg Q \vee \neg R, \neg P \vee S, P \vee S, Q \vee R, \neg Q \vee S, \neg Q \vee P, P, Q, \neg R, \neg P \vee \neg Q \vee R\}$

P	Q	R	S	α	Número de cláusulas satisfechas
Verdadero	***Falso***	***Verdadero***	***Falso***	***Falso***	***7***
Falso	*Falso*	*Verdadero*	*Falso*	*Falso*	*6*
Verdadero	*Verdadero*	*Verdadero*	*Falso*	*Falso*	*6*
Verdadero	*Falso*	*Falso*	*Falso*	*Falso*	*7*
Cierto	***Falso***	***Verdadero***	***Verdadero***	***Falso***	***8***
...					

⌘ WALKSAT

A veces realiza un camino aleatorio en vez de un criterio avaricioso.

- Si α es insatisfacible, WALKSAT nunca termina (si no se limita el número máximo de iteraciones).

Problemas SAT difíciles

Considerando conjuntos formados por 3 fbfs elegidas aleatoriamente con

m cláusulas

n símbolos diferentes,

Los problemas difíciles para WALKSAT y DPLL tienden a agruparse en torno a un "punto crítico" m/n = 4.3