Búsqueda e inferencia lógica

Estrategias de resolución

Contenidos

1. Introducción

Introducción

- Utilidad lógica simbólica
  - Representar problemas mediante FBF's
    - Conjunto de FBF's consistente (teoría)
  - Solucionar problemas de forma deductiva
    - Reglas de inferencia de la lógica
    - Proceso de búsqueda en el espacio de FBF's
- Dificultad: la aplicación no controlada de Reglas de Inferencia da lugar a un problema de explosión combinatoria
  - Crecimiento exponencial del nº de FBF's generadas

Ejemplo teoría

- Axiomas (propios de la teoría)
  - Todos los hombres son mortales
  - Sócrates es un hombre
- Teorema (se puede derivar de los axiomas propios)
  - Sócrates es mortal

Ejemplo representación en Lógica de Primer Orden

- Axiomas
  - Todos los hombres son mortales $\forall x(H(x) \supset M(x))$
  - Sócrates es un hombre $H(Socrates)$

- Teorema
  - Sócrates es mortal $M(Socrates)$

Reglas de inferencias

- Estructura

  antecedente $\rightarrow$ *consecuente*, donde

  antecedente o premisas: secuencia de patrones de FBF

  consecuente: secuencia de patrones de FBF

- Ejemplos:

  Modus Ponens $\alpha, \alpha \supset \beta \rightarrow \beta$

  Instanciación Universal $\forall x \alpha \rightarrow \beta$

Métodos uniformes de demostración

- Utilizan una única regla de inferencia
- Una forma de reducir la complejidad de la búsqueda
- Requieren transformar fórmulas a formato estándar

- Refutación por resolución
  - Para probar que $\Omega \vDash \alpha$, probar que $\Omega \cup \neg\alpha$ es inconsistente
  - única regla de inferencia: resolución

Dos aproximaciones básicas

- Sistemas de resolución
  - Transformar FBF’s a forma de cláusula
  - Aplicar resolución hasta generar cláusula vacía
  - Reducción complejidad sin limitar capacidad representación: selección de estrategias adecuadas

- Programación lógica
  - Sólo cláusula de Horn
  - Resolución SLD
  - Reducción complejidad limitando capacidad representación: cláusulas de Horn

2. Refutación por resolución

Refutación por resolución: procedimiento

- Sea T una teoría sólida y completa y t una FBF. Para probar $\exists\, T \vdash t$ mediante refutación por resolución:
  - Convertir los axiomas de T a FNC. Crear el conjunto S_0 como la conjunción de todas las cláusulas obtenidas
  - Negar t y convertir a FNC. Añadir las cláusulas obtenidas a S_0, obteniendo S
  - Repetir hasta obtener □ o no se generen nuevas cláusulas.
    - Seleccionar dos cláusulas que se puedan resolver, formando su resolvente
    - Si el resolvente no es □, añadir a S

Ejemplo transformación a cláusulas

Axiomas, LPO	Axiomas, clausulas
$\forall x(H(x) \supset M(x))$	$\neg H(x) \vee M(X)$
$H(Socrates)$	$H(Socrates)$

- Teorema
  - M(Socrates) — $M(Socrates)$

Refutación por resolución: S_0 y negación de *t*

- $S_0=\{\neg H(x) \vee M(x) , H(Socrates) \}$
- $\neg t = \neg M(Socrates)$

- $S =\{\neg H(x) \vee M(x) , H(Socrates), \neg M(Socrates)\}$

Buscar cláusula vacía (I)

$\neg H(x) \vee M(x)$ $\quad\quad$ $H(Socrates)$ $\quad\quad$ $\neg M(Socrates)$

$S=\{Socrates/x\}$

$M(Socrates)$

Buscar cláusula vacía (II)

$\neg H(x) \vee M(x)$ $\qquad$ $H(Socrates)$ $\qquad$ $\neg M(Socrates)\}$

$S=\{Socrates/x\}$

$M(Socrates)$

$\square$

Refutación por resolución: parada

- Lógica proposicional es decidible: siempre termina
  - Generando cláusula vacía
    - S inconsistente, $S_0 \cup \{\neg t\}$ inconsistente, $S_0 \vDash t$, $T \vDash t$ (si T completa $\exists\ T \vdash t$)
  - No se generan nuevas resolventes
    - S consistente, $S_0 \cup \{\neg t\}$ consistente, $S_0 \nvDash t$, $T \nvDash t$, (si T sólida, $\nexists\ T \vdash t$)

Refutación por resolución: parada

- Lógica de primer orden es semidecidible: el cómputo de nuevas resolventes puede no terminar, finalizando el proceso por consumo de recursos
  - Si el cómputo termina (parada), como en el caso proposicional
    - Generando cláusula vacía: S inconsistente
    - No se generan nuevas resolventes: S consistente
  - Si finaliza por consumo de recursos: no sabemos nada
    - S consistente, se pueden generar infinitas resolventes sin generar $\square$
    - S inconsistente, “podríamos” generar $\square$ asignando más recursos

3. Estrategias de resolución

Estrategias de Resolución

- Necesidad
  - la generación incontrolada de cláusulas hace que estas crezcan de forma exponencial
- Tipología
  - Simplificación: eliminan o reemplazan
  - Dirección: siguiente cláusula a considerar
  - Restricción: evitan generación de resolventes

Estrategia completa

- Una estrategia de resolución es completa (para la refutación) sii usada con una regla de inferencia completa (para la refutación) garantiza que encuentra una derivación de □ a partir de una forma clausulada inconsistente

- La regla de resolución es completa para la refutación

Estrategia de saturación por niveles

- Estrategia de dirección (similar bpa)

  Conjunto base: Conjunto S de todas las cláusulas de partida
  $S^0=S$. Si $k \in S$, k cláusula de nivel *0*
  $S^i=\{res(k_1, k_2) \,/\, k_1 \in (S^0 \cup S^1 \cup \ldots \ldots \cup S^{i-1}),\ k_2 \in S^{i-1}\}$
  Si $k \in S^i$, $k \notin S^{i-1}$, k cláusula de nivel *i-esimo*

- Estrategia de resolución por niveles: obtener primero todas las resolventes de nivel *i* antes de obtener una resolvente de nivel *i+1*

- COMPLETA e ineficiente

Estrategia de saturación por niveles (I)

- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$

Estrategia de saturación por niveles (II)

- S^0

1. $p \lor q$
2. $\neg p \lor q$
3. $p \lor \neg q$
4. $\neg p \lor \neg q$

Estrategia de saturación por niveles (III)

- S^0

1. $p \vee q$
2. $\neg p \vee q$
3. $p \vee \neg q$
4. $\neg p \vee \neg q$

- S^1

5. q de 1) y 2)
6. P de 1) y 3)
7. $q \vee \neg q$ *de 1) y 4)*
8. $p \vee \neg p$ *de 1) y 4)*
9. $q \vee \neg q$ *de 2) y 3)*
10. $\neg p \vee p$ *de 2 y 3)*
11. $\neg p$ *de 2 y 4)*
12. $\neg q$ *de 3 y 4)*

Estrategia de saturación por niveles (IV)

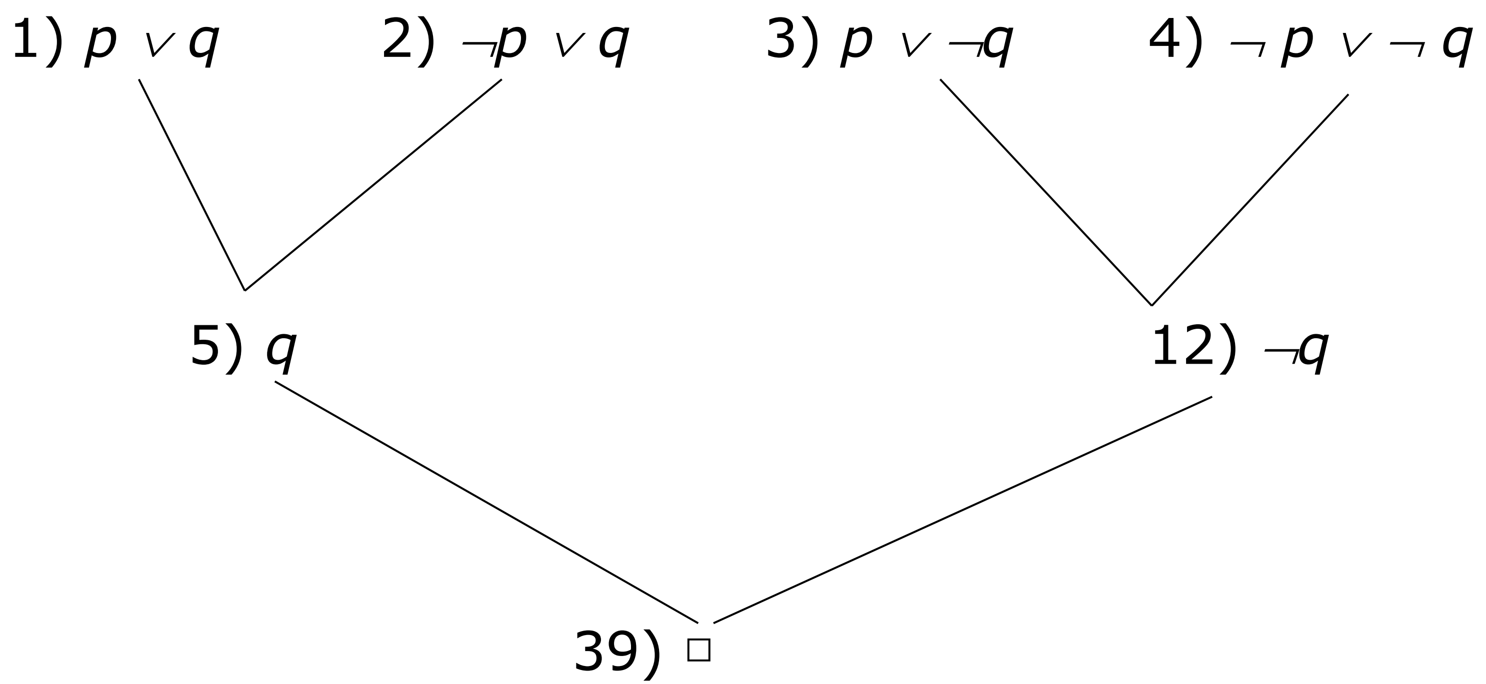
- S^1

5.	q	de 1) y 2)
6.	P	de 1) y 3)
7.	$q \vee \neg q$	*de 1) y 4)*
8.	$p \vee \neg p$	*de 1) y 4)*
9.	$q \vee \neg q$	*de 2) y 3)*
10.	$\neg p \vee p$	*de 2 y 3)*
11.	$\neg p$	*de 2 y 4)*
12.	$\neg q$	*de 3 y 4)*

- S^2

13.	$p \vee q$	de 1) y 7)
14.	.	
15.	.	
39.	□	de 5) y 12)

Completa e Ineficiente



Estrategias de simplificación

- Eliminación de literales puros
- Eliminación de tautologías
- Eliminación de cláusulas subsumidas
- Asociación de procedimientos

Eliminación de literales puros

- Def. literal puro
  - S forma clausulada, $k \in S$, $\lambda \in k$ es un literal puro sii $\nexists k' \in S$ con $\neg \mu \in k'$ y substituciones $s1$ y $s2$ / $\lambda\, s1 = \mu\, s2$

- Estrategia de eliminación de literales puros
  - eliminar todas las cláusulas que contengan literales puros
  - sólo al conjunto inicial

- COMPLETA

Eliminación de literales puros (I)

- $S = \{P(x) \vee Q(x), \neg P(x) \vee Q(x), P(x) \vee \neg Q(x), \neg P(x) \vee \neg Q(x), \neg P(x) \vee \neg Q(A)\}$

Eliminación de literales puros (II)

- $S = \{P(x) \vee Q(x), \neg P(x) \vee Q(x), P(x) \vee \neg Q(x), \neg P(x) \vee \neg Q(x), \neg P(x) \vee \neg Q(A)\}$

- No hay ningún literal puro

Eliminación de literales puros (III)

- $S = \{P(B) \vee Q(B), \neg P(B) \vee Q(B), P(B) \vee \neg Q(B), \neg P(B) \vee \neg Q(B), \neg P(B) \vee \neg Q(A)\}$

Eliminación de literales puros (IV)

- $S = \{P(B) \vee Q(B), \neg P(B) \vee Q(B), P(B) \vee \neg Q(B), \neg P(B) \vee \neg Q(B), \neg P(B) \vee \neg Q(A)\}$
- $\neg\, Q(A)$ es un literal puro
- Se genera un nuevo S:
- $S = \{P(B) \vee Q(B), \neg P(B) \vee Q(B), P(B) \vee \neg Q(B), \neg P(B) \vee \neg Q(B)\}$

Eliminación de tautologías

- Las cláusulas tautológicas no afectan a la satisfacibilidad
- Estrategia de eliminación de tautologías:
  - eliminar cláusulas tautológicas
  - iniciales y las que se generen

- COMPLETA

Eliminación de tautologías (I)

- S^0

  1. $p \lor q$
  2. $\neg p \lor q$
  3. $p \lor \neg q$
  4. $\neg p \lor \neg q$

- S^1

  5. q — de 1) y 2)
  6. P — de 1) y 3)
  7. $q \lor \neg q$ — *de 1) y 4)*
  8. $p \lor \neg p$ — *de 1) y 4)*
  9. $q \lor \neg q$ — *de 2) y 3)*
  10. $\neg p \lor p$ — *de 2 y 3)*
  11. $\neg p$ — *de 2 y 4)*
  12. $\neg q$ — *de 3 y 4)*

Eliminación de tautologías (II)

- S^0

1. $p \vee q$
2. $\neg p \vee q$
3. $p \vee \neg q$
4. $\neg p \vee \neg q$

- S^1

5. q de 1) y 2)
6. P de 1) y 3)
7. $\cancel{q \vee \neg q}$ *de 1) y 4)*
7. $\cancel{p \vee \neg p}$ *de 1) y 4)*
8. $\cancel{q \vee \neg q}$ *de 2) y 3)*
9. $\cancel{\neg p \vee p}$ *de 2 y 3)*
7. $\neg p$ *de 2 y 4)*
8. $\neg q$ *de 3 y 4)*

Eliminación de cláusulas subsumidas

- Def. subsunción

  k_1, k_2 cláusulas.

  k_1 subsume a k_2 *sii* $\exists$ substitución s / $k_1 s \subseteq k_2$

  k_2 cláusula subsumida

- Estrategia de eliminación de cláusulas subsumidas
  - Hacia delante: la resolvente puede ser subsumida
  - Hacia atrás: la resolvente puede subsumir

Ejemplos de cláusulas subsumidas

- $p \vee q$ subsume a $p \vee q \vee r$
- $P(A)$ subsume a $P(A) \vee Q(X)$ *(substitucion vacía)*
- $P(X)$ subsume a $P(A) \vee Q(X)$ $(s=\{A/X\})$
- $P(X)$ subsume a $P(A)$ $(s=\{A/X\})$

Eliminación de cláusulas subsumidas (I)

- S^0
  1. $p \vee q$
  2. $\neg p \vee q$
  3. $p \vee \neg q$
  4. $\neg p \vee \neg q$

- S^1
  5. q de 1) y 2)

Eliminación de cláusulas subsumidas (II)

- S^0
  1. $\cancel{p \vee q}$ subsumida por 5)
  2. $\cancel{\neg p \vee q}$ subsumida por 5)
  3. $p \vee \neg q$
  4. $\neg p \vee \neg q$

- S^1
  5. q de 1) y 2)

Eliminación de cláusulas subsumidas (II)

- S^0
  1. ~~$p \vee q$~~ subsumida por 5)
  2. ~~$\neg p \vee q$~~ subsumida por 5)
  3. $p \vee \neg q$
  4. $\neg p \vee \neg q$

- S^1
  5. q de 1) y 2)
  6. $\neg q$ de 3) y 4)

Eliminación de cláusulas subsumidas (III)

- S^0

  1. ~~$p \vee q$~~ subsumida por 5)
  2. ~~$\neg p \vee q$~~ subsumida por 5)
  3. ~~$p \vee \neg q$~~ subsumida por 6)
  4. ~~$\neg p \vee \neg q$~~ subsumida por 6)

- S^1

  5. q de 1) y 2)
  6. $\neg q$ de 3) y 4)

- S^2

  7. $\square$ *de 5) y 6)*

Eliminación de cláusulas subsumidas

- Completa con saturación por niveles. Puede no serlo con alguna estrategia de restricción

- MUY EFICIENTE: su aplicación suele ser imprescindible

Asociación de procedimientos

- Estrategia de simplificación/demoduladores
- Consiste en evaluar funciones o literales básicos sobre un dominio (interpretación parcial)
- Afecta a la satisfacibilidad pues estamos fijando una interpretación parcial
- NO es completa

Evaluación de funciones

- Asociar un símbolo de función con un procedimiento cuya evaluación devuelva un elemento del dominio
- Evaluar particularizaciones básicas del término funcional
- Reemplazar el término funcional por el elemento de dominio

$K=P(x) \lor Q(suma(3,5), y)$

se transforma en

$K=P(x) \lor Q(8, y)$ (con la interpretación habitual de la suma)

Evaluación de literales

- Asociar un símbolo de predicado con un procedimiento, cuya evaluación devuelva un valor de verdad
- Evaluar particularizaciones básicas del literal
- Si el literal se evalúa a *T*: eliminar la cláusula
- Si el literal se evalúa a *F*: eliminar el literal de la cláusula

  $K=P(x) \vee MAYOR(3,5)$

  se transforma en

  $K=P(x)$ (asumiendo $V(MAYOR(3,5))=F$)

Evaluación de literales

- *S={¬PAR(x) ∨ IMPAR(suc(x)), ¬IMPAR(x) ∨ PAR(suc(x)), PAR(X) ∨ IMPAR(X), PAR(4) ∨ F(4)}*

- Interpretación parcial:
  - D=N
  - $PAR^I=\{d \in D \ / \ resto(d, 2)=0\}$, $IMPAR^I= D-PAR^I$
  - $suc^I(d)=d+1$

- Evaluar particularizaciones básicas del literal: V(PAR(4))=T

- *S'={¬PAR(x) ∨ IMPAR(suc(x)), ¬IMPAR(x) ∨ PAR(suc(x)), PAR(X) ∨ IMPAR(X)}*

- Todos los modelos de S que mantengan la interpretación parcial también son modelos de S' (y viceversa)

Estrategias de restricción

- Conjunto soporte
- Resolución lineal
- Resolución por entradas
- Resolución unitaria

Estrategia del conjunto soporte

- Def. Conjunto soporte
  - S forma clausulada y $T \subset S$, $T \neq S$, $\varnothing$
  - T se denomina conjunto soporte de S

- Def. conjunto soporte nivel i-esimo
  - $T_0 = T$, T conjunto soporte de S. T_0 se denomina conjunto soporte de nivel 0
  - T_i, conjunto soporte de nivel i-esimo: conjunto de $res(k_l, k_m)$ con:
    1. $\exists\, j \,/\, k_j \in T_i\text{-}1$ (o factor de cláusula de $T_i\text{-}1$)
    2. (la cláusula que no cumple 1) $\in S$ (o factor de cláusula de S)

Estrategia del conjunto soporte

- Def. T-soporte de una cláusula

  S forma clausulada, *T* conjunto soporte, *k* cláusula

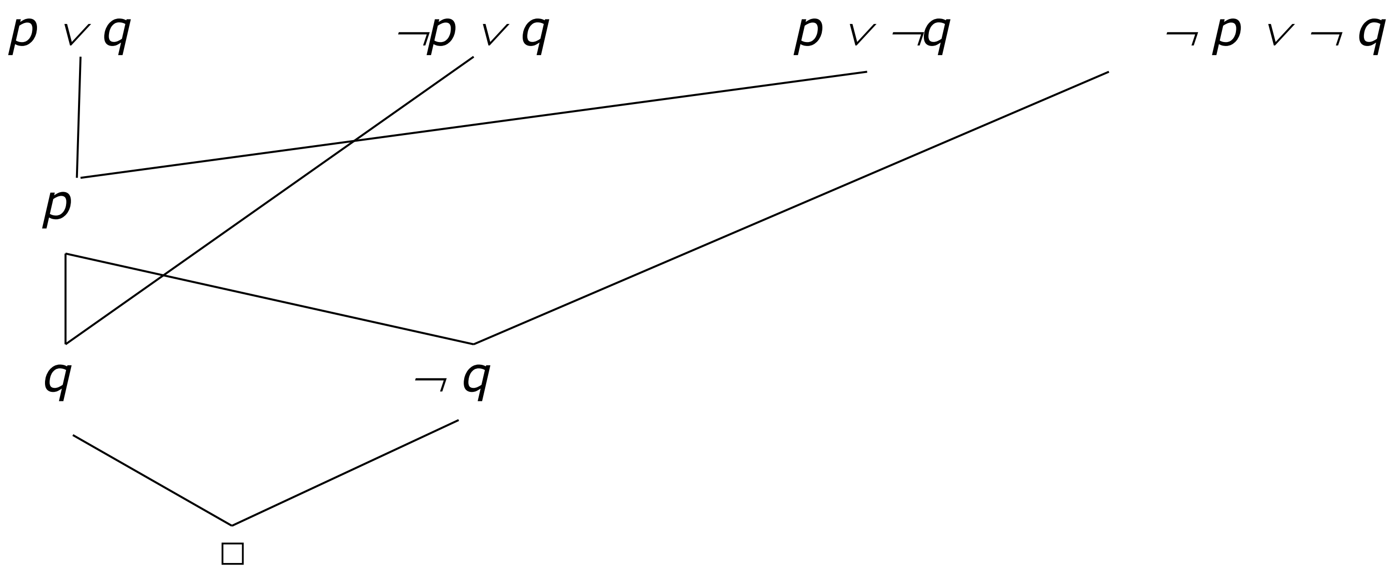
  K tiene *T-soporte sii* $k \in T_i$, $i \geq 0$

- Estrategia del conjunto soporte

  S conjunto base, *T* conjunto soporte. La estrategia del conjunto soporte solo permite obtener cláusulas que tengan *T-soporte*

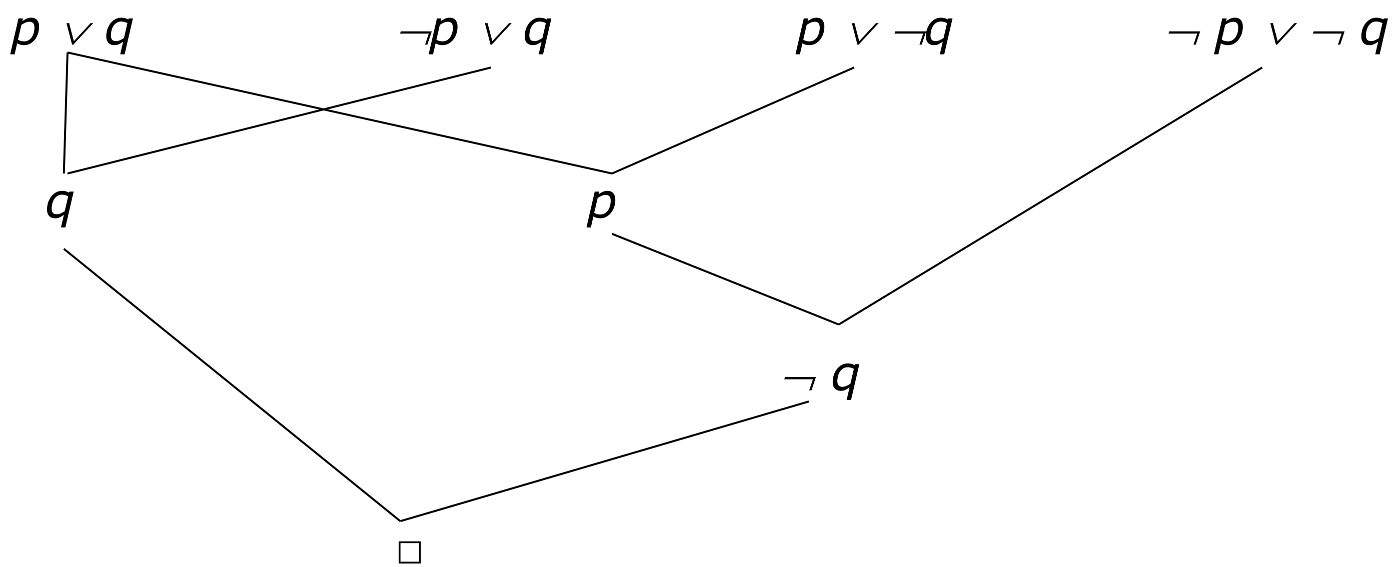
Estrategia del conjunto soporte

- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$
- $T = \{p \vee q\}$



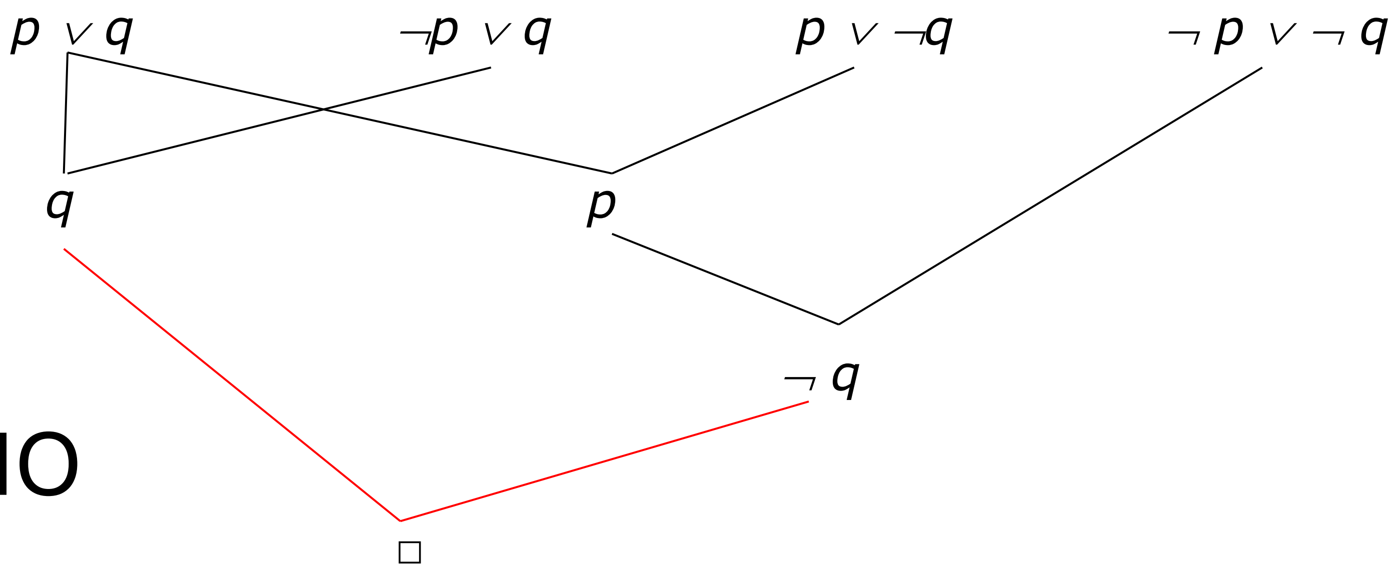
¿Estrategia del conjunto soporte?

- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$
- $T = \{p \vee q\}$



¿Estrategia del conjunto soporte?

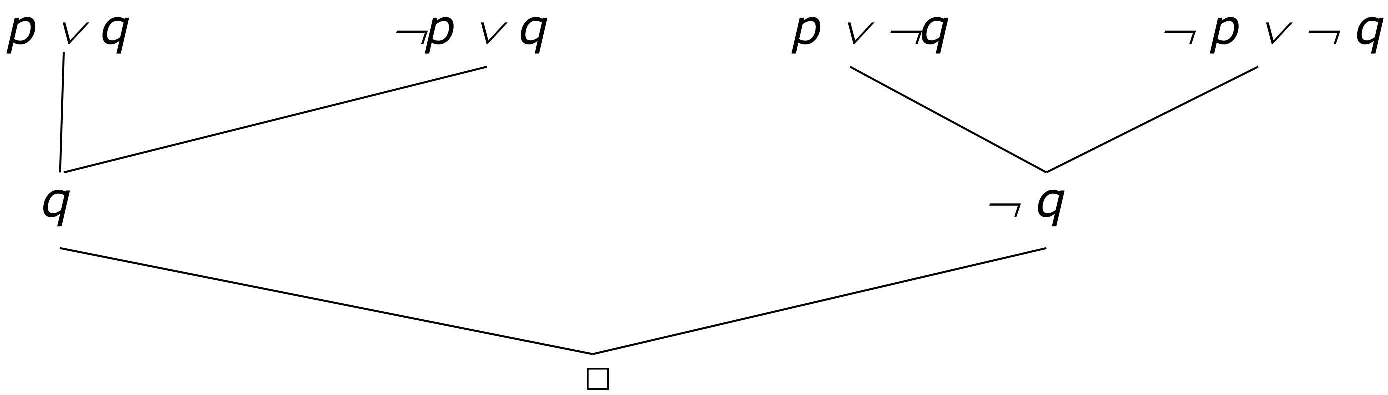
- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$
- $T = \{p \vee q\}$



NO

Estrategia del conjunto soporte

- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$
- $T = \{p \vee q, \neg p \vee \neg q\}$



Teorema conjunto soporte

- Sea *S* forma clausulada inconsistente y *T* un conjunto soporte de *S* / *S*-*T* sea consistente.

  $\exists\ S \vdash_r \square$ utilizando la estrategia del conjunto soporte, con *T* como conjunto soporte

- Elección habitual de *T*
  - *Th(A)* teoría de axiomas propios *A*, *t* teorema
  - *S*: cláusulas de *Th(A)* y $\neg t$, inconsistente si *t* teorema
  - *T*: cláusulas que provienen de *t*
  - *S*-*T*: cláusulas de *Th(A)*, normalmente consistente

Resolución lineal

- S forma clausulada, $kc_0 \in S$ cláusula central de partida

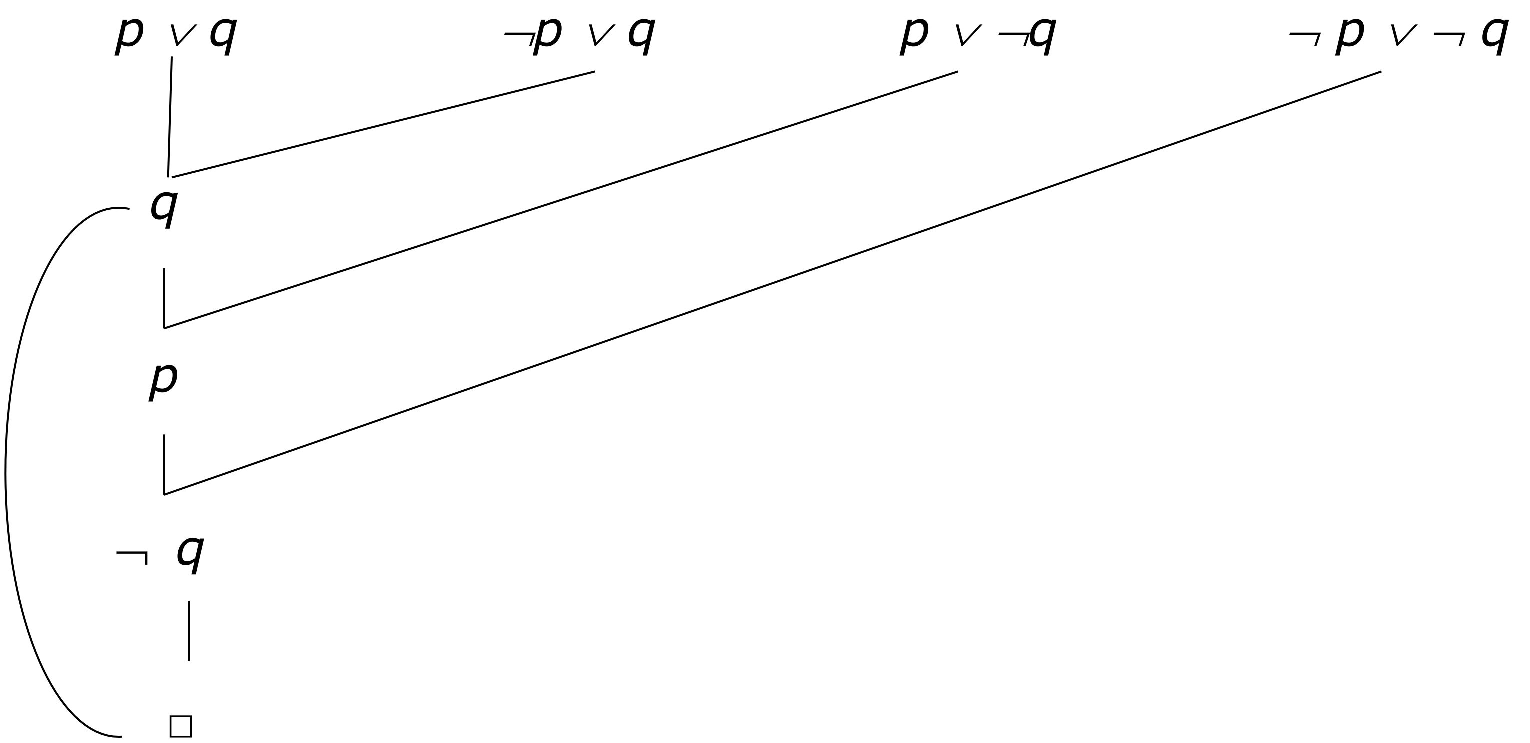
  La estrategia lineal sólo permite obtener como resolventes cláusulas centrales kc_{i+1}, con:

  1. $kc_{i+1}=res(kc_i, B_i)$
  2. $B_i \in S$ ó $B_i = kc_j$ con $j<i$ (o factor)

  B_i : cláusula lateral

Resolución lineal

- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$
- $KC_0 = p \vee q$



Teorema complitud resolución lineal

- Sea S forma clausulada inconsistente y $kc_0 \in S$, de modo que S-$\{kc_0\}$ sea consistente.

  $\exists\, S \vdash_r \square$ utilizando la estrategia de resolución lineal, con kc_0 como cláusula central de partida.

- Elección kc_0
  - Como conjunto soporte si la negación del teorema da lugar a una única cláusula.

Resolución por entradas

- Def. cláusula de entrada: cláusulas del conjunto base, S
- Def. resolvente de entrada: resolvente con al menos un padre cláusula de entrada

- Estrategia de resolución por entradas: sólo permite obtener resolventes de entrada

- NO es completa

Resolución por entradas

- $S = \{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$

$p \qquad \neg p \vee q \qquad p \vee \neg q \qquad \neg p \vee \neg q$

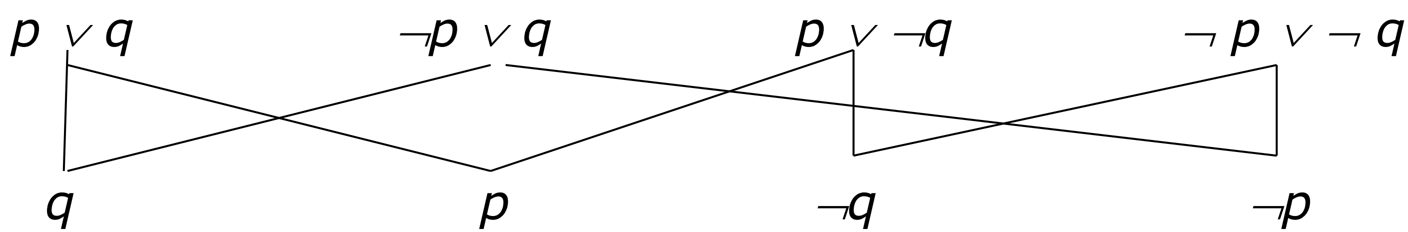
q

$\neg p$

$\square$

Resolución por entradas: no es completa

- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$



Resolución unitaria

- Def. resolvente unitario: resolvente en el que al menos una de las cláusulas padres es unitaria.
- Estrategia de resolución unitaria: sólo permite obtener resolventes unitarios.
- NO es completa

Resolución unitaria

- $S = \{p\ ,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\}$

$p \qquad\qquad \neg p \vee q \qquad\qquad p \vee \neg q \qquad\qquad \neg p \vee \neg q$

q

$\neg p$

$\square$

También por entradas y conjunto soporte

Resolución unitaria: no es completa

- $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$
- $KC_0 = p \vee q$

$p \vee q$ $\qquad$ $\neg p \vee q$ $\qquad$ $p \vee \neg q$ $\qquad$ $\neg p \vee \neg q$

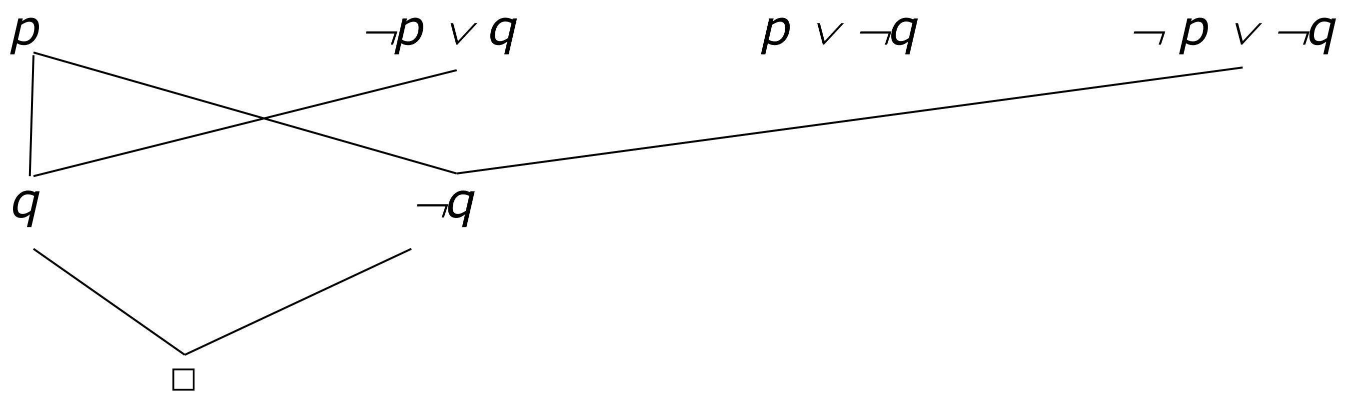
Teorema equivalencia resolución unitaria y por entradas

Sea S forma clausulada.

$\exists$ $S \vdash_r \square$ utilizando la estrategia de resolución unitaria *sii* $\exists$ $S \vdash_r \square$ utilizando la estrategia de resolución por entradas

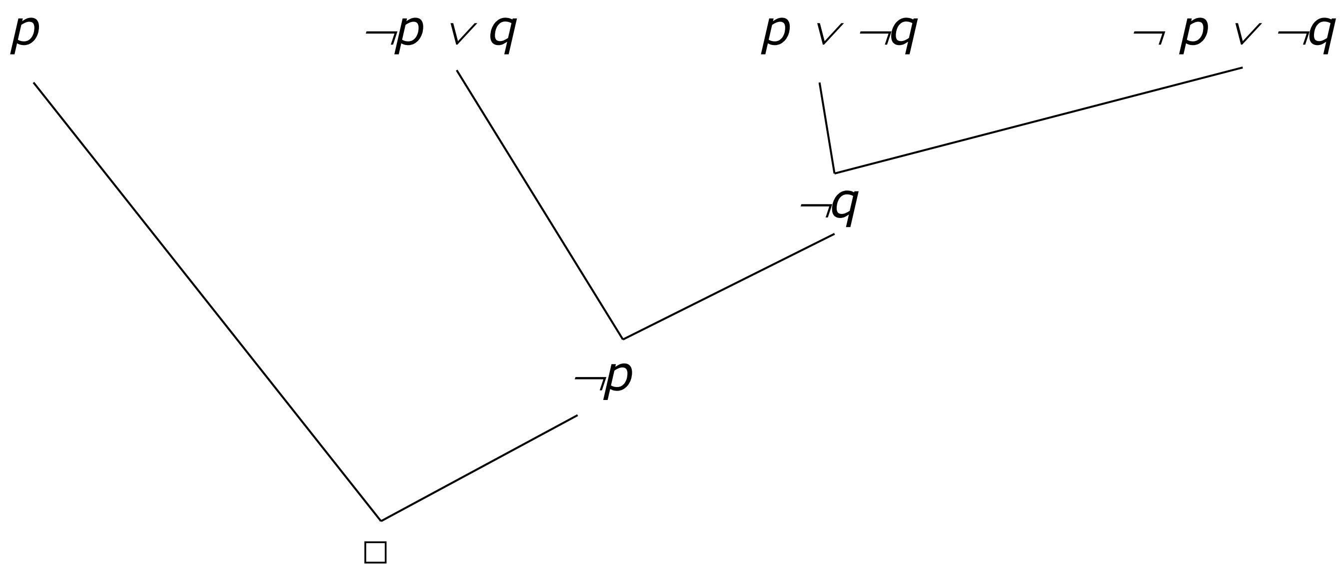
Resolución unitaria, no por entradas

- $S = \{p\ ,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\}$



Resolución por entradas, no unitaria

- $S = \{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$



Cláusulas de Horn

- Def. Cláusula de Horn (definidas): cláusula que tiene, a lo sumo, un literal positivo.

- Las cláusulas de Horn se pueden interpretar como implicaciones, con el literal positivo a la derecha del símbolo implica:

$P(x) \vee \neg Q(x) \vee \neg R(y)$	$\forall x \forall y\ (Q(x) \wedge R(y) \supset P(x))$
$P(x)$	$\forall x\ (\supset P(x))$
$\neg Q(x) \vee \neg R(y)$	$\forall x \forall y\ (Q(x) \wedge R(y) \supset)$

Teorema complitud resolución por entradas (unitaria)

- Sea *H* una forma clausulada cuyos elementos son cláusulas de Horn.

- *H* es inconsistente *sii* $\exists S \vdash_r \square$ utilizando la estrategia de resolución por entradas (unitaria)

4. Procedimiento de extracción de respuesta

Procedimiento de extracción de respuesta

- Def. Extracción de respuesta mediante refutación por resolución

  proceso de encontrar los elementos del dominio que hacen cierto el teorema a demostrar, mediante una prueba de refutación por resolución

Preguntas

- En general, cualquier FBF
- Por motivos prácticos, sentencias en FNP con
  - Matriz: conjunción de literales
  - Prefijo: sólo cuantificadores existenciales

  $\exists x \exists y (P(x) \wedge Q(x, y))$
- Interpretación
  - La demostración se puede interpretar como la pregunta: ¿Existen substituciones de variables que hagan cierto el teorema?
- Propiedad
  - La negación de estos teoremas da lugar a una única cláusula

  $\neg \exists x \exists y (P(x) \wedge Q(x, y)) = \forall x \forall y (\neg P(x) \vee \neg Q(x, y))$

Literal respuesta

- Def. Literal respuesta

Sea P pregunta y $x_1, x_2,\dots \dots,x_n$ las variables que ocurren en P.

Un literal respuesta para P es:

$$RES(x_1, x_2,\dots \dots,x_n)$$

Obtención de la respuesta

- Si *P* es una pregunta y $RES(x_1, x_2, \ldots \ldots, x_n)$ su literal respuesta, la respuesta se obtiene
  1. negando *P* y transformándolo a cláusulas
  2. formando la disyunción de las cláusulas obtenidas en 1 con $RES(x_1, x_2, \ldots \ldots, x_n)$
  3. buscando derivaciones de cláusulas que solo contengan literales respuestas

Ejemplo de pregunta I

- Axiomas, LPO
  - $\forall x(H(x) \supset M(x))$
  - $H(Socrates)$

Axiomas, clausulas

$\neg H(x) \vee M(X)$

$H(Socrates)$

- Teorema: Pregunta
  - $\exists M(x)$

$\neg M(x)$

Obtención de la respuesta I

$\neg H(x) \vee M(x)$ $\quad H(Socrates)$ $\quad \neg M(y) \vee RES(y)$

$S1=\{Socrates/x\}$

$M(Socrates)$

$S2=\{Socrates/y\}$

$RES(Socrates)$

Ejemplo de pregunta II

- Axiomas, LPO
  - $\forall x(H(x) \supset M(x))$
  - $H(Socrates)$
- Teorema: Pregunta
  - M(Socrates)

Axiomas, clausulas

$\neg H(x) \vee M(X)$

$H(Socrates)$

$\neg M(Socrates)$

Obtención de la respuesta II

$\neg H(x) \vee M(x)$ $\quad H(Socrates)$ $\quad \neg M(Socrates) \vee RES$

$S=\{Socrates/x\}$

$M(Socrates)$

RES

Propiedades de la respuesta

- Puede contener más de un literal
- No es única
  - Depende de la derivación que la produce

Respuesta: más de un literal

- España o Alemania ganará el mundial.
- ¿Quién ganará el mundial?

GANARA(España) $\vee$ *GANARA(Alemania)*

$\exists x$*GANARA(X)*

GANARA(España) $\vee$ *GANARA(Alemania)* $\neg$*GANARA(x)* $\vee$ *RES(x)*

S1={España/x}

GANARA(Alemania) $\vee$ *RES(España)*

S2={Alemania/x}

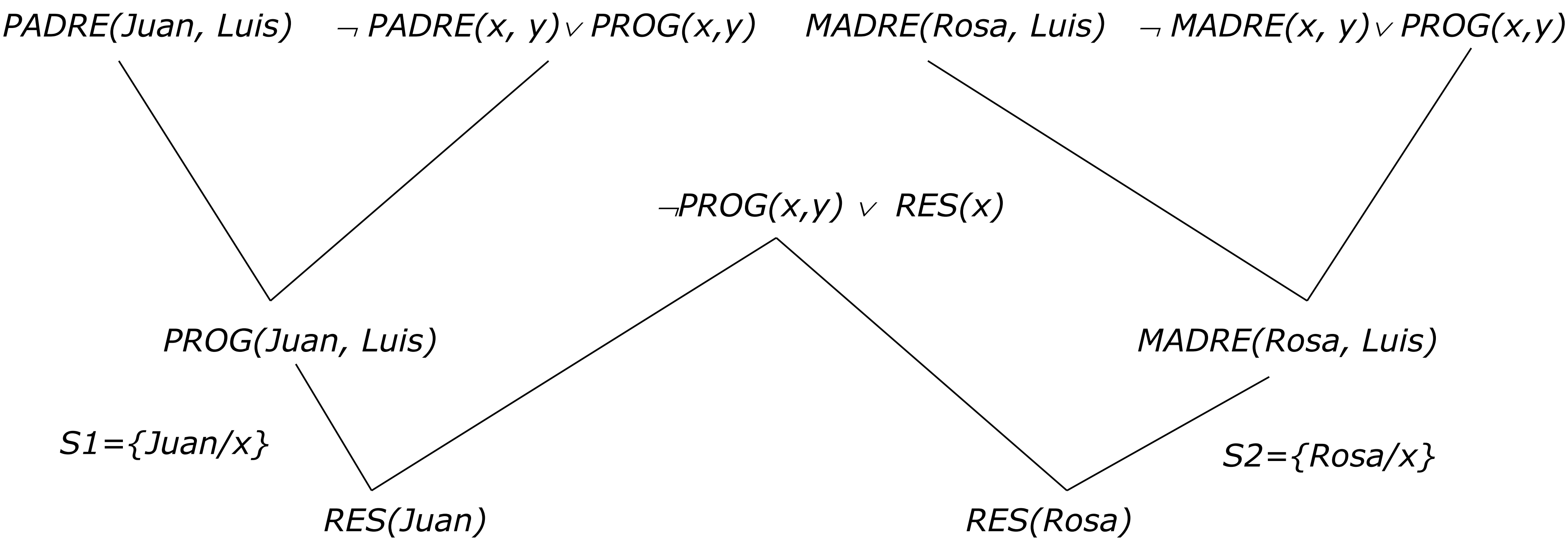
RES(España) $\vee$ *RES(Alemania)*

Respuesta: depende de la refutación encontrada

- Teoría:
  - Juan es el padre de Luis.
  - Rosa es la madre de Luis.
  - El padre de una persona es uno de sus progenitores.
  - La madre de una persona es uno de sus progenitores

- Pregunta:
  - ¿Quién es el progenitor de Luis?

Respuesta:
depende de la refutación encontrada



5. Demostradores de teoremas

Demostradores de teoremas

- Programas que dada *Th(AP)* y FBF *t*, intentan comprobar si $\exists\, AP \vdash t$
- Generalmente, aceptan FBF de LPO como entrada
- Generalmente basados en refutación por resolución
- Estrategias de control para limitar búsqueda

PTTP (*Prolog Technology Theorem Prover*)

- Búsqueda: descenso iterativo, completa (en vez de bpp)
  - La inferencia es completa con la regla de resolución lineal y por entradas
- Negación lógica: se implementa una rutina para probar *P* y otra para probar *notP*
- (Re)Introduce chequeo de ocurrencias en la unificación

OTTER (*Organized Techniques for Theorem-proving and Effective Research*)

- Refutación por resolución
- Uno de los primeros y más populares
  - sucesor: prover9
- Libre disposición
  - http://www.cs.unm.edu/~mccune/otter/
- Utilizado en
  - Investigación matemática
  - Verificación de hardware

Entrada de OTTER

- Hechos importantes sobre el dominio (SOS, Set of Support)
- Conocimiento sobre el problema: axiomas de utilidad (Usables)
- Demoduladores: reglas de reescritura
- Parámetros y cláusulas que definen la estrategia de control

Procedimiento OTTER

```
Procedimiento OTTER(SOS, Usables)
      entrada: SOS, Usables
      repetir
            cláusula <- elemento de SOS de menor peso
            llevar cláusula de SOS a Usables
            Procesar (Inferir (cláusula,  Usables), SOS)
      hasta SOS=∅ ó se ha encontrado refutación
      end OTTER
```

Procedimiento OTTER

- Inferir (cláusula, Usables)
  - Resuelve cláusula con todas las de Usables
  - Filtra cláusulas
- Procesar (resolventes, SOS)
  - Estrategias simplificación
  - Llevar cláusulas a SOS según pesos
  - Comprobar presencia de cláusula unitaria y su complementario