Lógica prop.

Sistemas Inteligentes I

Tema 5. Agentes Lógicos

José A. Montenegro Montes
monte@lcc.uma.es

Resumen

- Introducción
- Entorno
- Lógica
  - Lógica proposicional
- Fundamentos
- Demostración Teoremas
- Lógica Primer Orden

Introducción

Agentes lógicos

- Los humanos conocen hechos, y lo que saben les ayuda a actuar
  - Los procesos de razonamiento trabajan con representaciones internas del conocimiento
- La Inteligencia Artificial construye agentes basados en el conocimiento que también son capaces de razonar
- CSPs introduce la idea de representar los estados cómo asignaciones de valores a variables
  - Permite problemas independiente del dominio y algoritmos más eficientes.

Agentes lógicos

- Continuamos con la idea introducida por CSPs
- Establecemos la lógica cómo una clase general de representaciones para definir agentes basados en el conocimiento (KbA).
- KbA aceptan tareas que son objetivos descritos de forma explícita.
  - Pueden aprender nuevos conocimientos sobre el entorno
  - Adaptare a cambios en el entorno modificando su conocimiento

Agentes basados en Conocimiento

- **Base de conocimiento** (KB) es el elemento principal.
  - Es un conjunto de **sentencias** (sentences).
- **Lenguaje representación del conocimiento** es utilizado para expresar la sentencias, que son afirmaciones sobre el mundo (entorno)
- **Axiomas** son sentencias que no son derivadas de otras sentencias.
- Procedimiento para añadir y consultar sentencias a la base de conocimiento (**TELL y ASK**)
  - Operaciones implican **inferencias, derivar** nuevas sentencias de antiguas

Agentes basados en Conocimiento

- **Funcionamiento agente común,** damos una percepción a KB y obtenemos una acción.

function KB-AGENT($percept$) **returns** an $action$
  persistent: KB, a knowledge base
    t, a counter, initially 0, indicating time

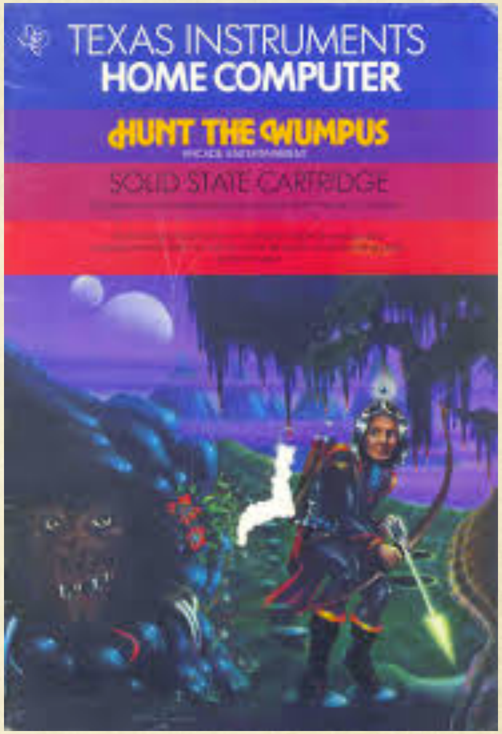
  TELL(KB, MAKE-PERCEPT-SENTENCE($percept$, t))
  $action \leftarrow$ ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE($action$, t))
  $t \leftarrow t + 1$
  return $action$

Entorno

Wumpus World

Wumpus World

- Cueva con una serie de habitaciones conectados por pasadizos.
- Bestia (wumpus) está en la cueva esperando comerse a quien entre en la cueva.
- El agente solo tiene una flecha para disparar a la bestia.
- Algunas habitaciones tienen huecos profundos.
- Lo único positivo de este entorno es que puedes encontrar oro en algunas habitaciones.



Wumpus World

- **Medida de rendimiento**:
  - +1000 Salir de la cueva con oro
  - -1000 Caer en un hueco o que el wumpus coma al agente
  - -1 Por cada acción realiza
  - -10 Por utilizar la flecha
  - El juego finaliza cuando el agente muere o cuando el agente sale de la cueva.
- **Entorno**:
  - Una mapa de 4 x4 cuadrículas
  - El agente siempre comienza en la cuadricula [1,1]
  - El oro y el wumpus son establecidos de forma aleatoria.
  - Cada cuadricula puede ser un hueco con probabilidad 0.2

Wumpus World

- **Actuadores**:
  - Agente puede moverse:
    - Hacia delante
    - Girar 90 a la izquierda
    - Girar 90 a la derecha.
  - Choca con una pared no se mueve.

  - Acciones:
    - ***Grab***, coger oro que este en la misma posición que el agente.
    - ***Shoot***, dispara una flecha en línea directa y la flecha continua hasta matar al wumpus o chocar con una pared.
    - ***Climb***, salir de la cueva, pero solamente desde cuadrícula [1,1].

Wumpus World

- **Sensores**: Cinco sensores:
  - Hedor (stench), en la cuadricula que está el wumpus y en las adyacentes
  - Brisa (breeze), en las casillas adyacentes a un hueco
  - Brillo (glitter), en la casilla donde está el oro
  - Golpe (bump), cuando choca con una pared.
  - Grito (scream), cuando matan al wumpus.

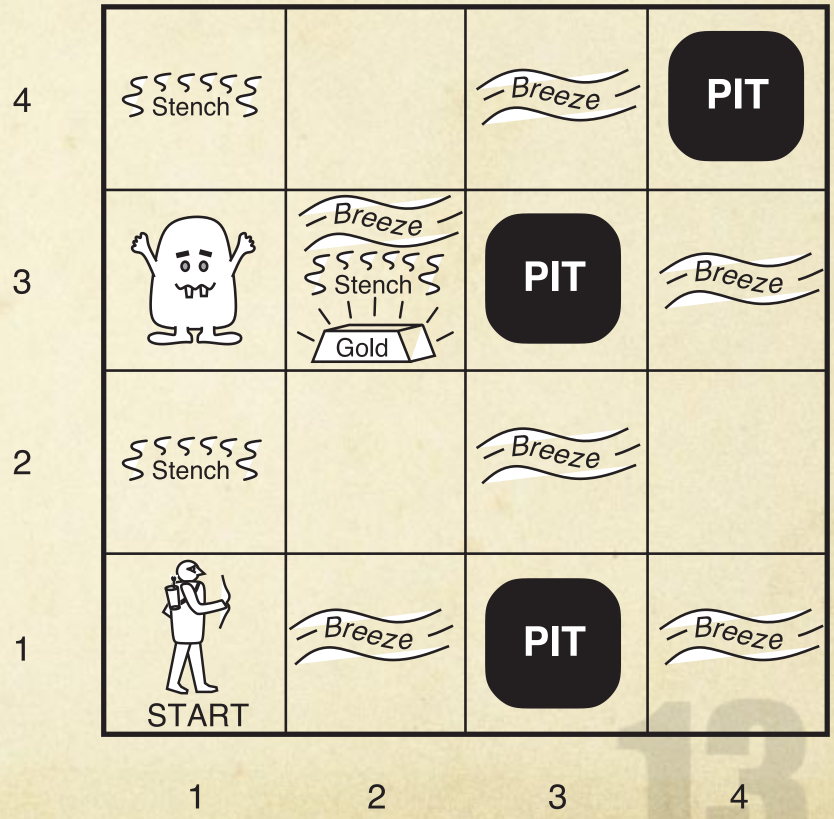
El agente obtiene la información cómo una lista de cinco símbolos. Por ejemplo:

[Stench, Breeze, None, None, None]

El agente debe escoger si salir a salvo con las manos vacías o arriesgarse a recoger el oro. Aprox. 21% entornos el oro está rodeado de huecos.

Wumpus World

- KbA Wumpus para explorar el entorno.
- Lenguaje informal para representar el conocimiento, escribiendo ***símbolos*** en las cuadrículas.
- Inicialmente KB del agente contiene las reglas del entorno y que en la cuadrícula [1,1] esta a salvo.



Wumpus World

- $\boxed{\text{A}}$ = *Agent*
- **B** = *Breeze*
- **G** = *Glitter, Gold*
- **OK** = *Safe square*
- **P** = *Pit*
- **S** = *Stench*
- **V** = *Visited*
- **W** = *Wumpus*

- La primera percepción del agente es

  [None, None, None, None, None]

- Intuimos que las casillas [1,2] y [2,1] no tienen peligro.
- Son marcadas con OK, y forman parte de la KB del agente.

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 **OK**	2,2	3,2	4,2
1,1 $\boxed{\text{A}}$ **OK**	2,1 **OK**	3,1	4,1

Wumpus World

- **A** = *Agent*
- **B** = *Breeze*
- **G** = *Glitter, Gold*
- **OK** = *Safe square*
- **P** = *Pit*
- **S** = *Stench*
- **V** = *Visited*
- **W** = *Wumpus*

- Agente se mueve al [2,1]

  [None, Breeze, None, None, None]

- Tiene que existir un hueco cerca [2,2] o [3,1]
- Agente volvería a una cuadrícula que este a salvo.

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 $\boxed{A}$ B OK	3,1 P?	4,1

Wumpus World

A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

- Agente se mueve al [1,2]

  [Stench, None, None, None, None]

- La nueva percepción hace pensar varias situaciones:
  - Wumpus estará en [1,3] o [2,2] pero [2,2] no puede estar ya que antes no obtuvo información.
  - En el [2,2] no hay un hueco, ya que no recibo la brisa, con lo cual el hueco tiene que estar en 3,1,
  - Y finalmente la casilla [2,2] SAFE

1,4	2,4	3,4	4,4
1,3 **W!**	2,3	3,3	4,3
1,2 **A** **S** **OK**	2,2 **OK**	3,2	4,2
1,1 **V** **OK**	2,1 **B** **V** **OK**	3,1 **P!**	4,1

Wumpus World

- **A** = *Agent*
- **B** = *Breeze*
- **G** = *Glitter, Gold*
- **OK** = *Safe square*
- **P** = *Pit*
- **S** = *Stench*
- **V** = *Visited*
- **W** = *Wumpus*

- Agente se mueve al [2,2] y [2,3]

  [Stench, Breeze, Glitter, None, None]

- Agente coge el oro y debe volver a casa.
- Agente llega a una conclusión desde la información disponible.
  - Las conclusiones son correctas si la información disponible era correcta

1,4	2,4 **P?**	3,4	4,4
1,3 **W!**	2,3 **A** **S G B**	3,3 **P?**	4,3
1,2 **S V OK**	2,2 **V OK**	3,2	4,2
1,1 **V OK**	2,1 **B V OK**	3,1 **P!**	4,1

Lógica

Lógica

- KB está formada de sentencias.
- **Sintaxis**: Define la correcta representación de las sentencias en un lenguaje dado.
  - x+y=4 sentencia bien formada
  - x4y+= sentencia no cumple sintaxis

Lógica

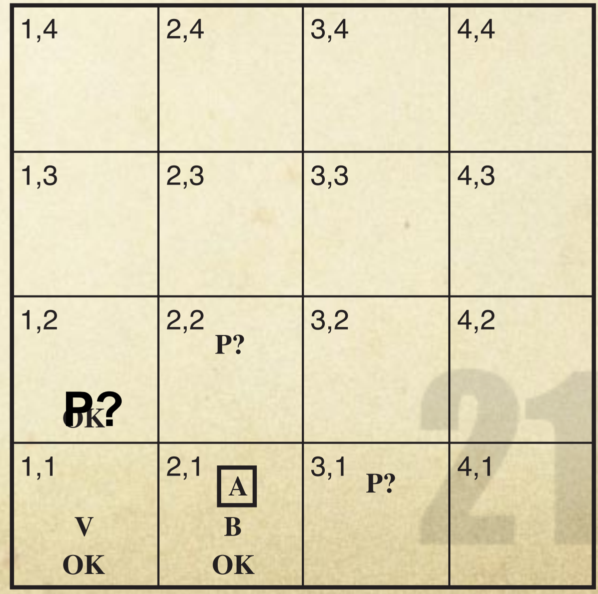
- **Semántica**: Significado de las sentencias. Definen la verdad de cada sentencia dependiendo de un mundo posible (**modelo**).
  - x+y = 4 es verdad en un mundo donde x e y =2.
  - Los posibles modelos son todas las posibles asignaciones a las variables x e y.

  - Si la sentencia α es verdad en el modelo m,
    - m satisface α o
    - m es el modelo de α

  - M(α) el conjunto de todos los modelos de α

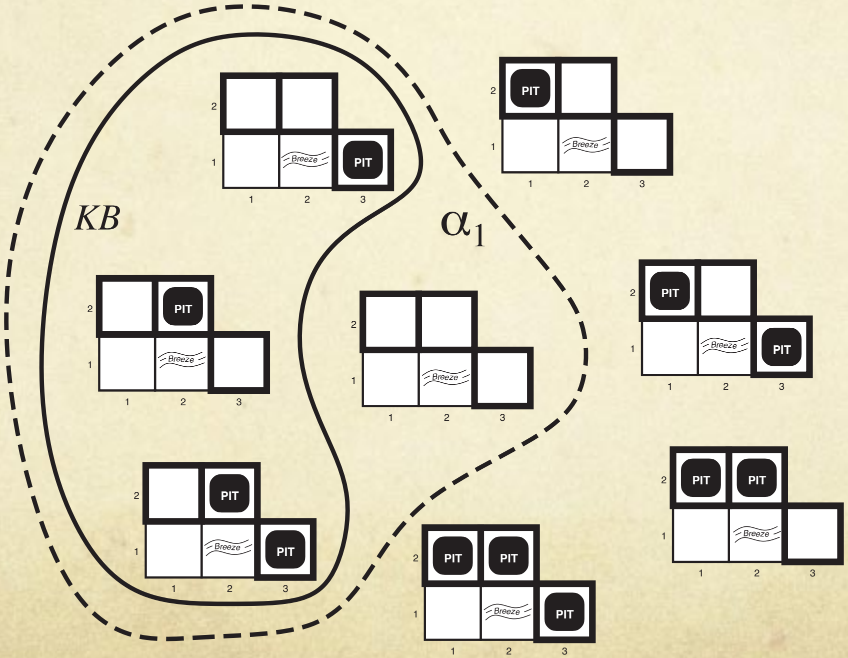
Lógica

- **Semántica**:
  - $\alpha \models \beta$ Consecuencia lógica (infiere) entre sentencias. La idea es que una sentencia "sigue lógicamente" de otra sentencia.
  - α **se infiere de** β sii en todo modelo en el que α es verdadera, β también es verdadera.
- **Wumpus world**:
  - El agente no detecta nada en [1,1] y una brisa en [2,1]
  - Estas percepciones, junto a las reglas son KB.
  - Agente está pensando si las tres casillas [1,2],[2,2] y [3,1] tienen huecos.
  - Hay 2^3=8 modelos posibles.

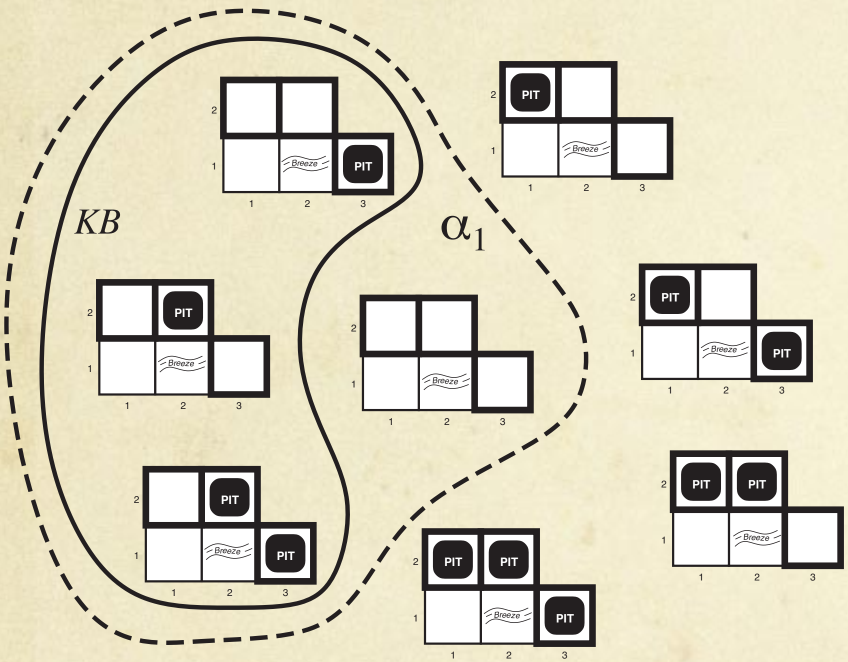


Lógica

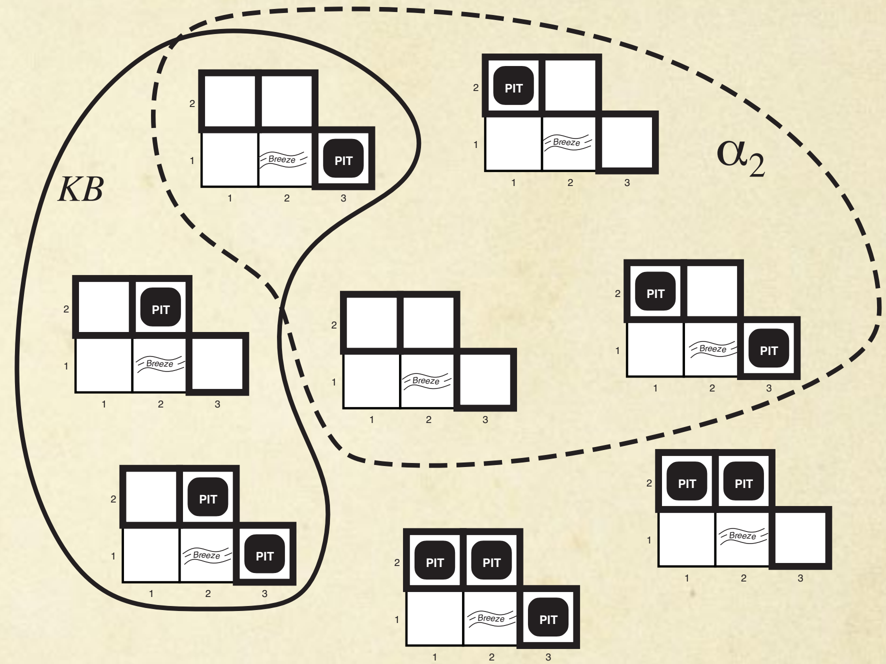
- KB es **falsa** en los modelos que contradicen lo que el agente conoce.
  - KB es falsa en cualquier modelo el cual [1,2] tenga un hueco, ya que no percibimos brisa en el [1,1].
  - Solo tenemos tres modelos verdaderos.



Wumpus World

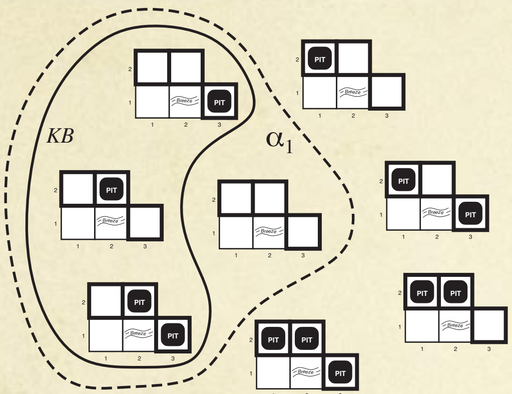


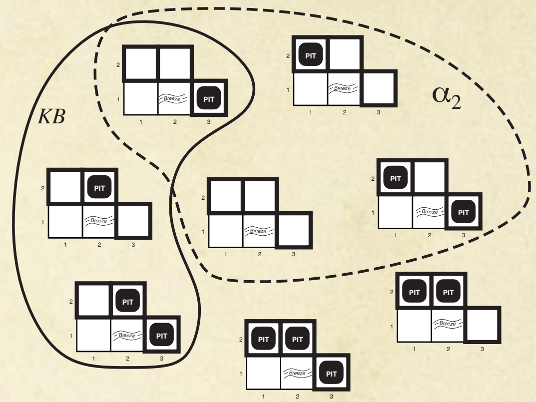
α_1 = No hay hueco en [1,2]



α_2 = No hay hueco en [2,2]

Wumpus World





En cada modelo que KB es verdadero, α_1 es también verdadero, por tanto KB $\models \alpha_1$: no hay hueco en [1,2]

En algunos modelos KB es verdadero, α_2 es falso, por tanto, KB $\not\models \alpha_2$: no podemos determinar si hay hueco en [2,2]

Model Checking: Enumero todos los modelos posibles para verificar que α es verdad en todos los modelos los cuales KB es verdadero.

Lógica Proposicional

Sintaxis

Sintaxis (I)

- La sintaxis de la lógica proposicional define las fórmulas bien formadas
- Las fórmulas atómicas (también llamadas átomos) consisten de un solo símbolo de proposición: P, Q, $Rains$, W_{13}(Wumpus cuadrícula 1,3), …
  - Dos símbolos especiales: *True* y *False*
- Las fórmulas compuestas se obtienen de fórmulas más sencillas empleando los paréntesis y las conectivas lógicas
- A continuación se presentan las cinco conectivas que usaremos, en **orden de precedencia**

Sintaxis (II)

- $\neg$ (no). Un literal es, o bien una fórmula atómica (literal positivo), o bien su negación (literal negativo)
- $\wedge$ (y). Una fórmula cuya conectiva de nivel más alto es $\wedge$, se denomina conjunción
- $\vee$ (o). Una fórmula cuya conectiva de nivel más alto es $\vee$, se denomina disyunción
- $\longrightarrow$ (implica). Una fórmula del tipo $\alpha \longrightarrow \beta$ se llama implicación, donde α es la premisa o antecedente, y β es la conclusión o consecuencia
- $\longleftrightarrow$ (si y sólo si). Una fórmula cuya conectiva de nivel más alto es $\longleftrightarrow$, se denomina bicondicional

Fundamentos

Semántica

Semántica (I)

- La semántica define las reglas para determinar la verdad de una sentencia con respecto a un modelo particular
- En la lógica proposicional, un modelo fija el **valor de verdad** (verdadero o falso) de todos los símbolos de proposición:

$$m_1=\{P_{1,2} = \text{false}, P_{2,2}= \text{false}, P_{3,1}=\text{true}\}$$

- La semántica debe especificar como calcular el **valor de verdad** de cualquier sentencia, dado un modelo. Todas las sentencias son construidas mediante sentencia atómicas y cinco conectores.

Semántica (II)

- *Sentencias atómicas:*
  - *True* es verdadero en todo modelo, y *False* es falso en todo modelo
  - El valor de verdad de los demás símbolos lo especifica el modelo, p.ej. en m_1 $P_{1,2}$ es falso.

- *Sentencias complejas, con subsentencias P y Q en cualquier modelo*:
  - $\neg P$ es verdadero sii P es falso en m
  - $P \wedge Q$ es verdadero sii tanto P como Q son verdaderos en m
  - $P \vee Q$ es verdadero sii P o bien Q son verdaderos en m
  - $P \longrightarrow Q$ es verdadero a menos que P sea verdadero y Q sea falso en m
  - $P \leftrightarrow Q$ es verdadero sii P y Q son ambos verdaderos o ambos falsos en m

Semántica (III)

Tablas de Verdad

A	B	$A \wedge B$
V	V	V
V	F	F
F	V	F
F	F	F

A	B	$A \vee B$
V	V	V
V	F	V
F	V	V
F	F	F

A	$\neg A$
V	F
F	V

A	B	$A \rightarrow B$
V	V	V
V	F	F
F	V	V
F	F	V

A	B	$A \leftrightarrow B$
V	V	V
V	F	F
F	V	F
F	F	V

Semántica (IV)

- Si una sentencia α es verdadera en un modelo *m*, decimos que ***m* satisface** α
- β **se infiere de** α ($\alpha \models \beta$) sii en todo modelo en el que α es verdadera, β también es verdadera.
- Una sentencia es válida sii es verdadera en todos los modelos; en tal caso decimos que es una tautología
- Una sentencia es **satisfacible** sii es verdadera en algún modelo
- Una sentencia es **insatisfacible** si no es satisfacible
- Por último, se cumple que $\alpha \models \beta$ sii $(\alpha \wedge \neg \beta)$ es insatisfacible

Ejemplo (Ejercicio 1)

- Demuestra que las siguientes fórmulas bien formadas son tautologías:
  - $[P \wedge (P \longrightarrow Q)] \longrightarrow Q$

P	Q	$P \Rightarrow Q$	$P \wedge (P \Rightarrow Q)$	$[P \wedge (P \Rightarrow Q)] \Rightarrow Q$
V	V	V	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	V

  - $(P \longrightarrow Q) \longleftrightarrow (\neg Q \longrightarrow \neg P)$

P	Q	$\neg Q$	$\neg P$	$P \Rightarrow Q$	$\neg Q \Rightarrow \neg P$	$(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P)$
V	V	F	F	V	V	V
V	F	V	F	F	F	V
F	V	F	V	V	V	V
F	F	V	V	V	V	V

Fundamentos

Base de conocimiento

Bases de conocimiento

- Base de conocimientos para Wumpus world:
  - $P_{x,y}$ es verdad si hay un hueco en [x,y]
  - $W_{x,y}$ es verdad si hay un wumpus en [x,y], vivo o muerto
  - $B_{x,y}$ es verdad si el agente percibe una brisa en [x,y]
  - $S_{x,y}$ es verdad si el agente percibe una brillo en [x,y]

- Queremos saber si no hay hueco en [1,2] ($\neg P_{1,2}$)
  - No hay hueco en [1,1]
    - R_1: $\neg P_{1,1}$
  - Hay brisa sii hay un hueco en una cuadrícula vecina. Es necesario realizarlo para cada cuadrícula
    - R_2: $B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})$
    - R_3: $B_{2,1} \leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
  - Las anteriores reglas son verdad en todos los mundos wumpus.
  - Ahora incluimos las situaciones específicas después de visitar las dos primeras cuadrículas:
    - R_4: $\neg B_{1,1}$
    - R_5: $B_{2,1}$

Fundamentos

Procedimiento simple inferencia

Procedimiento simple inferencia

- Nuestro objetivo es decidir si KB |= α para alguna sentencia α.
  - *¬ $P_{1,2}$ se infiere de nuestra KB.*
- *Model-checking:* Enumerara todos los modelos y verificar que α es verdadero en cada modelo en el cual KB es verdadero.
  - 2^7=128 modelos, en los cuales 3 KB son verdaderos.
  - ¬ $P_{1,2}$ son verdaderos en esos 3, por tanto no hay hueco [1,2].
  - $P_{2,2}$ verdadero en dos de los tres, no puedo decir si hay hueco en [2,2]

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	false	false	false	false	false	false	true	true	true	true	false	false
false	false	false	false	false	false	true	true	true	false	true	false	false
false	true	false	false	false	false	true	true	true	true	true	true	**true**
false	true	false	false	false	true	false	false	true	true	true	true	**true**
false	true	false	false	false	true	true	true	true	true	true	true	**true**

Demostración Teoremas

Demostración por resolución

- Una regla de inferencia toma varias sentencias y produce otra sentencias que puede inferirse de ellas
- Una demostración es una secuencia de sentencias obtenida por aplicación de reglas de inferencia a partir de una KB
- Sólo consideraremos una regla de inferencia, la regla de resolución
  - La resolución es correcta, es decir, nunca produce una fórmula que no se infiera de la KB
  - También es completa, es decir, cuando se combina con cualquier algoritmo de búsqueda completo, es capaz de alcanzar cualquier fórmula que pueda deducirse de la KB

La regla de inferencia de resolución

- La resolución toma dos cláusulas (disyunciones de literales) tales que hay un literal l_i en la primera cláusula que es la negación de un literal m_j de la segunda cláusula, o sea, l_i y m_j son literales complementarios.

C1. $\neg Q$

C2. $\neg R \vee Q$

C3. $\neg R$ Resolver C1 con C2

C1. $P_{1,1} \vee P_{3,1}$

C2. $\neg P_{1,1} \vee \neg P_{2,2}$

C3. $P_{3,1} \vee \neg P_{2,2}$ Resolver C1 con C2

La regla de inferencia de resolución

- Produce una cláusula con todos los literales de las dos cláusulas originales excepto los dos literales complementarios.

$$\frac{l_1 \vee \ldots \vee l_k, \quad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

- Las apariciones repetidas de literales son también eliminadas de la cláusula resultante.

C1. $\neg P \vee Q$

C2. $P \vee Q$

C3. Q Resolver C1 con C2

Demostración Teoremas

Forma normal conjuntiva

Forma normal conjuntiva

- La resolución sólo se puede aplicar a cláusulas
- Una sentencia que es una conjunción de cláusulas se dice que está en forma normal conjuntiva (*conjunctive normal form*, CNF)
- Toda fórmula de la lógica proposicional es lógicamente equivalente a una conjunción de cláusulas
  - Un algoritmo para convertir a CNF sería

Conversión a CNF

1. Eliminar $\leftrightarrow$ reemplazando
   - $\alpha \leftrightarrow \beta$ por $(\alpha \longrightarrow \beta) \wedge (\beta \rightarrow \alpha)$
2. Eliminar $\longrightarrow$ reemplazando
   - $\alpha \rightarrow \beta$ por $\neg \alpha \vee \beta$
3. Mover $\neg$ hacia dentro aplicando repetidamente:
   - $\neg(\neg \alpha) \equiv \alpha$
   - $\neg(\alpha \wedge \beta) \equiv \neg \alpha \vee \neg \beta$
   - $\neg(\alpha \vee \beta) \equiv \neg \alpha \wedge \neg \beta$
4. Aplicar la distributividad de $\vee$ respecto a $\wedge$:
   - $\alpha \vee (\beta \wedge z) \equiv (\alpha \vee \beta) \wedge (\alpha \vee z)$

Ejemplos

$\boldsymbol{p \longrightarrow ((q \longrightarrow r) \vee \neg\, s)\ ;}$

$\equiv \quad p \longrightarrow ((\neg q \vee r) \vee \neg\, s);$

$\equiv \quad \neg\, p \vee ((\neg q \vee r) \vee \neg\, s)$

$\boldsymbol{\neg(\neg p \wedge (q \wedge \neg\, (r \wedge s)));}$

$\equiv \quad \neg\neg p \vee \neg\, (q \wedge \neg\, (r \wedge s));$

$\equiv \quad p \vee (\neg\, q \vee \neg\neg\, (r \wedge s));$

$\equiv \quad p \vee (\neg\, q \vee (r \wedge s));$

$\equiv \quad p \vee ((\neg\, q \vee r) \wedge (\neg\, q \vee s));$

$\equiv \quad (p \vee \neg\, q \vee r) \wedge (p \vee \neg\, q \vee s));$

Ejemplos

$(p \land q) \lor (p \land \neg q)$

$\equiv \quad ((p \land q) \lor p) \land ((p \land q) \lor \neg q)$

$\equiv \quad ((p \lor p) \land (q \lor p)) \land ((p \lor \neg q) \land (q \lor \neg q))$

$((p \land q) \lor (r \land s)) \lor (\neg q \land (p \lor t))$

Ejercicio

Demostración Teoremas

Un algoritmo de resolución

Un algoritmo de resolución

- Nuestro objetivo es demostrar que KB $\models \alpha$. Lo haremos por reducción al absurdo, o sea, demostraremos que KB $\wedge \neg \alpha$ es insatisfacible

1. Primero convertimos KB $\wedge \neg \alpha$ a CNF
2. Después aplicamos la regla de resolución repetidamente

- Hay dos posibles resultados:
  - **No** se pueden añadir más cláusulas, lo que significa que α no se infiere de KB
  - Se produce la **cláusula vacía**, lo que significa que α se infiere de KB

Ejemplo

- Queremos saber si no hay hueco en [1,2] ¬ ($P_{1,2}$)
  - R_1: $\neg P_{1,1}$
  - R_2: $B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})$
  - R_3: $B_{2,1} \leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
  - R_4: $\neg B_{1,1}$
  - R_5: $B_{2,1}$

- $KB = R_2 \wedge R_4 = (B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$

  $(B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})) = (B_{1,1} \longrightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \longrightarrow B_{1,1})$

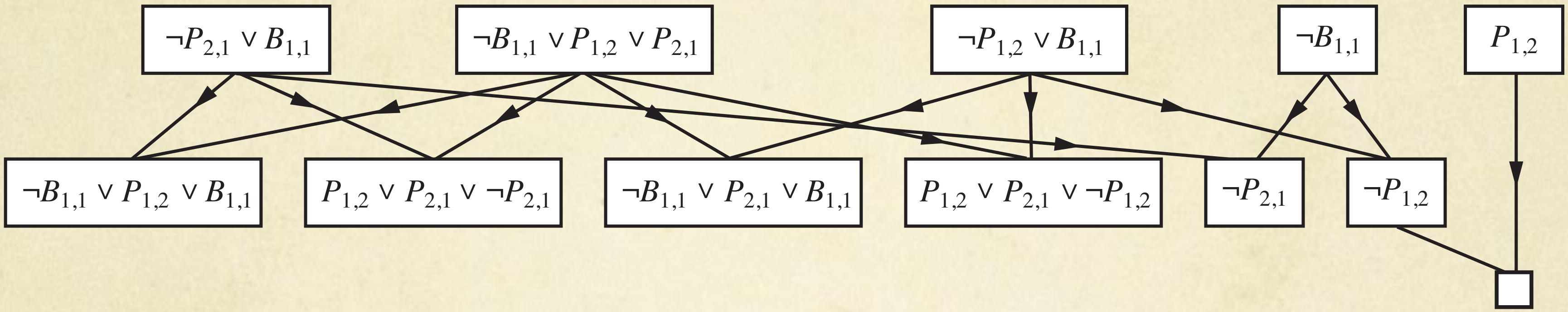
  $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

  $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

  $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

- $\alpha = \neg P_{1,2}$

Ejemplo



function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  $clauses \leftarrow$ the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
  $new \leftarrow \{\ \}$
  loop do
    for each pair of clauses C_i, C_j **in** $clauses$ **do**
      $resolvents \leftarrow$ PL-RESOLVE(C_i, C_j)
      if $resolvents$ contains the empty clause **then return** *true*
      $new \leftarrow new \cup resolvents$
    if $new \subseteq clauses$ **then return** *false*
    $clauses \leftarrow clauses \cup new$

Ejemplo I

- Una base de conocimiento sencilla:
  - Regla$_1$: $Wet \leftrightarrow (Rain \lor Flooding)$
  - Regla$_2$: $Hot \leftrightarrow (Summer \lor Sunny \lor Fire)$
  - Hecho$_1$: $\neg Summer$
  - Hecho$_2$: $\neg Wet$
  - Hecho$_3$: Hot

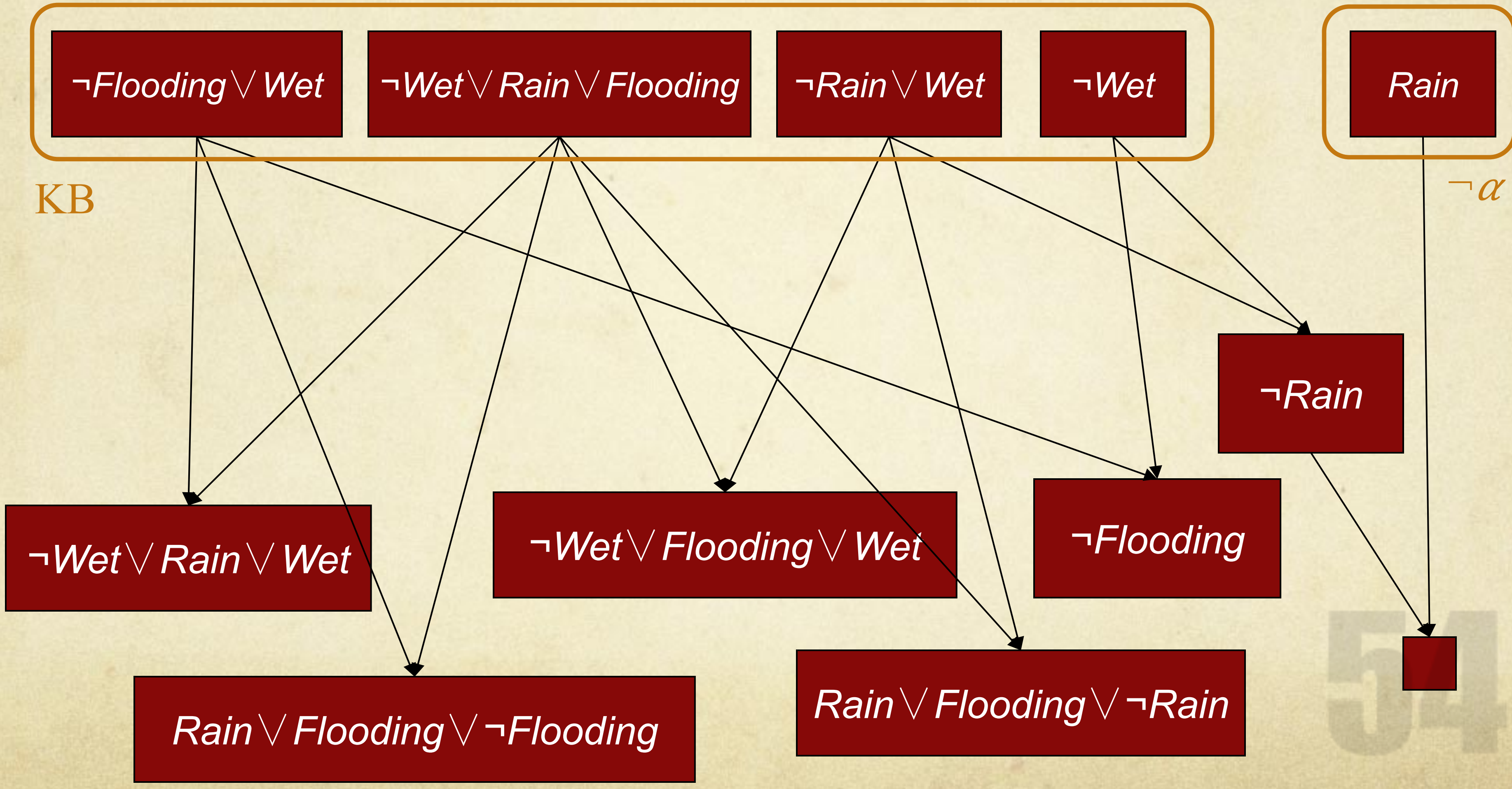
Ejemplo I

- Restringimos nuestra KB a Regla_1 y Hecho_2
  - Regla_1: $Wet \leftrightarrow (Rain \vee Flooding)$
  - Hecho_2: $\neg Wet$
- Queremos demostrar $\alpha = \neg Rain$
- Primero mostramos la conversión de KB a CNF

Ejemplo I

- $[\, Wet \leftrightarrow (Rain \lor Flooding)\,] \land \neg Wet$
- $[\, Wet \longrightarrow (Rain \lor Flooding)\,] \land [\, (Rain \lor Flooding) \longrightarrow Wet\,] \land \neg Wet$
- $[\, \neg Wet \lor Rain \lor Flooding\,] \land [\neg(Rain \lor Flooding) \lor Wet\,] \land \neg Wet$
- $[\, \neg Wet \lor Rain \lor Flooding\,] \land [\, (\neg Rain \land \neg Flooding) \lor Wet\,] \land \neg Wet$
- $(\, \neg Wet \lor Rain \lor Flooding\,) \land (\neg Rain \lor Wet\,) \land (\neg Flooding \lor Wet\,) \land \neg Wet$

Ejemplo I


¬Flooding ∨ Wet
¬Wet ∨ Rain ∨ Flooding
¬Rain ∨ Wet
¬Wet
Rain
KB
¬α
¬Rain
¬Wet ∨ Rain ∨ Wet
¬Wet ∨ Flooding ∨ Wet
¬Flooding
Rain ∨ Flooding ∨ ¬Flooding
Rain ∨ Flooding ∨ ¬Rain

Ejercicio 7.7

Sea un vocabulario con solamente cuatro proposiciones, A, B, C, y D. ¿Cuántos modelos hay para las siguientes fórmulas?

a. $B \vee C$

b. $\neg A \vee \neg B \vee \neg C \vee \neg D$

c. $(A \longrightarrow B) \wedge A \wedge \neg B \wedge C \wedge D$

Ejercicio 7.7

$2^4=16$ modelos

a. $B \vee C$

Es falso cuando B y C son falsos, solo ocurre en 4 modelos, por lo que 16-4 = 12 modelos.

b. $\neg A \vee \neg B \vee \neg C \vee \neg D$

Falso cuando $A \wedge \mathbf{B} \wedge C \wedge D$ solo ocurre 1 modelo, por lo que 16-1=15

c. $(A \longrightarrow B) \wedge A \wedge \neg B \wedge C \wedge D$

$(\neg A \vee B) \wedge A \wedge \neg B \wedge C \wedge D$

Falso si $(A \wedge \neg B)$ 0 modelos

A	B	C	D
F	F	F	F
F	F	F	T
F	F	T	F
F	F	T	T
F	T	F	F
F	T	F	T
F	T	T	F
F	T	T	T
T	F	F	F
T	F	F	T
T	F	T	F
T	F	T	T
T	T	F	F
T	T	F	T
T	T	T	T
T	T	T	T

Ejercicio 1

$\neg Q \wedge (R \longrightarrow Q)$

$\neg R \longrightarrow P$

$\models$

$\neg R$

Ejercicio 1

$\neg Q \wedge (R \longrightarrow Q)$

$\neg R \longrightarrow P$

$\models$

$\neg R$

C1. $\neg$ Q

C2. $\neg$ R $\vee$ Q

C3. R $\vee$ P

C4. R

C5. $\neg$R Resolver C1 con C2

C6. False Resolver C4 con C5

Ejercicio 1

$\neg Q \wedge (R \longrightarrow Q)$

$\neg R \longrightarrow P$

$\models$

$\neg R$

QRP	$\neg Q \wedge (R \longrightarrow Q)$	$\neg R \longrightarrow P$	$\wedge$	$\neg R$
000	1	0	0	1
001	1	1	1	1
010	0	1	0	0
011	0	1	0	0
100	0	0	0	1
101	0	1	0	1
110	0	1	0	0
111	0	1	0	0

Ejercicio 2

$P \leftrightarrow T$

$(T \longrightarrow \neg S) \leftrightarrow Q$

$\neg P$

$\models$

Q

Ejercicio 2

$P \leftrightarrow T$

$(T \longrightarrow \neg S) \leftrightarrow Q$

$\neg P$

$\models$

Q

C1. $\neg P \vee T$
C2. $P \vee \neg T$
C3. $T \vee Q$
C4. $S \vee Q$
C5. $\neg T \vee \neg S \vee \neg Q$
C6. $\neg P$
C7. $\neg Q$

C8. $\neg T$	Resuelvo C2 con C6
C9. Q	Resuelvo C3 con C8
C10. *False*	Resuelvo C7 con C9

Ejercicio 2

$P \leftrightarrow T$

$(T \longrightarrow \neg S) \leftrightarrow Q$

$\neg P$

$|=$

Q

P T S Q	$P \leftrightarrow T$	$(T \longrightarrow \neg S) \leftrightarrow Q$	$\neg P$	$\wedge$	Q
0000	1	0	1	0	0
0001	1	1	1	1	1
0010	1	0	1	0	0
0011	1	1	1	1	1
0100	0	0	1	0	0
0101	0	1	1	0	1
0110	0	1	1	0	0
0111	0	0	1	0	1
1000	0	0	0	0	0
1001	0	1	0	0	1
1010	0	0	0	0	0
1011	0	1	0	0	1
1100	1	0	0	0	0
1101	1	1	0	0	1
1110	1	1	0	0	0
1111	1	0	0	0	1

Ejercicio 3

$(P \vee Q) \leftrightarrow (R \wedge S)$
$P \longrightarrow Q$
$P \wedge S$

$|=$

R

Ejercicio 3

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

$P \longrightarrow Q$

$P \wedge S$

$\models$

R

P	Q	R	S	$A = P \vee Q$	$B = R \wedge S$	$A \leftrightarrow B$	$P \longrightarrow Q$	$P \wedge S$
0	0	0	0	0	0	1	1	0
0	0	0	1	0	0	1	1	0
0	0	1	0	0	0	1	1	0
0	0	1	1	0	1	0	1	0
0	1	0	0	1	0	0	1	0
0	1	0	1	1	0	0	1	0
0	1	1	0	1	0	0	1	0
0	1	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0	0
1	0	0	1	1	0	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	1	1	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	0	1	0
1	1	1	1	1	1	1	1	1

Ejercicio 3

$(P \vee Q) \leftrightarrow (R \wedge S)$
$P \longrightarrow Q$
$P \wedge S$

$|=$

R

C1. $\neg P \vee R$
C2. $\neg P \vee S$
C3. $\neg Q \vee R$
C4. $\neg Q \vee S$
C5. $\neg R \vee \neg S \vee P \vee Q$
C6. $\neg P \vee Q$
C7. P
C8. S
C9. $\neg R$

C10. R Resuelvo C1 con C7
C11. $False$ Resuelvo C9 con C10

Ejemplos

$$((\boldsymbol{p} \wedge \boldsymbol{q}) \vee (\boldsymbol{r} \wedge \boldsymbol{s})) \vee (\neg \boldsymbol{q} \wedge (\boldsymbol{p} \vee \boldsymbol{t}))$$

$$\equiv \; (((p \wedge q) \vee r) \wedge ((p \wedge q) \vee s)) \vee (\neg q \wedge (p \vee t))$$

$$\equiv \; ((p \vee r) \wedge (q \vee r) \wedge (p \vee s) \wedge (q \vee s)) \vee (\neg q \wedge (p \vee t))$$

$$\equiv \; ((p \vee r) \vee (\neg q \wedge (p \vee t)) \wedge$$
$$((q \vee r) \vee (\neg q \wedge (p \vee t)) \wedge$$
$$((p \vee s) \vee (\neg q \wedge (p \vee t)) \wedge$$
$$((q \vee s) \vee (\neg q \wedge (p \vee t))$$

$$\equiv \; (p \vee r \vee \neg q) \wedge (p \vee r \vee p \vee t)) \wedge$$
$$(q \vee r \vee \neg q) \wedge (q \vee r \vee p \vee t) \wedge$$
$$(p \vee s \vee \neg q) \wedge (p \vee s \vee p \vee t) \wedge$$
$$(q \vee s \vee \neg q) \wedge (q \vee s \vee p \vee t)$$

Ejemplos

$$((p \wedge q) \vee (r \wedge s)) \vee (\neg q \wedge (p \vee t))$$

$$\equiv \quad (p \vee r \vee \neg q) \wedge (p \vee r \vee p \vee t)) \wedge$$
$$(q \vee r \vee \neg q) \wedge (q \vee r \vee p \vee t) \wedge$$
$$(p \vee s \vee \neg q) \wedge (p \vee s \vee p \vee t) \wedge$$
$$(q \vee s \vee \neg q) \wedge (q \vee s \vee p \vee t)$$

$$\equiv \quad (p \vee r \vee \neg q) \wedge (p \vee r \vee t) \wedge (q \vee r \vee p \vee t) \wedge$$
$$(p \vee s \vee \neg q) \wedge (s \vee p \vee t) \wedge (q \vee s \vee p \vee t)$$

Introducción

Lógica primer orden

Generalidades

- La lógica proposicional es demasiado sencilla para representar el conocimiento en entornos complejos
- La lógica de primer orden toma prestadas ideas de los lenguajes naturales a la vez que evita sus ambigüedades
  - Se construye sobre los objetos y las relaciones entre ellos
  - Supone que dichas relaciones o se cumplen o no se cumplen entre los objetos

Modelos

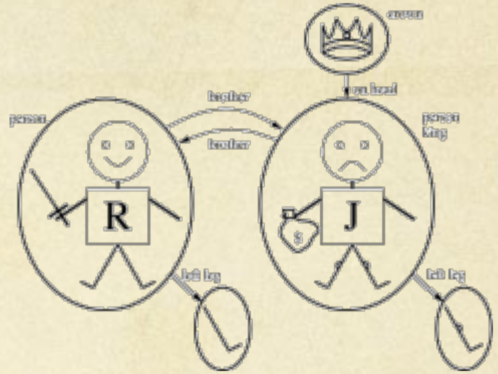
- Un modelo de la lógica de primer orden tiene los siguientes componentes:
  - Un dominio, que es el conjunto de objetos o elementos del dominio que contiene.
  - Un conjunto de relaciones entre objetos. Una relación es un conjunto de tuplas de objetos que están relacionados.

- **Ejemplo:**
  - **Objetos:** El rey Ricardo Corazón de León; su hermano menor, el malvado rey Juan; y una corona (en total 3 objetos)
  - **Relaciones:**
    - Hermandad={<Ricardo Corazón de León, Rey Juan>, <Rey Juan, Ricardo Corazón de León>}
    - Sobre la cabeza={<La corona, Rey Juan>}

Símbolos e interpretaciones

- Hay dos tipos de símbolos:
  - Símbolos de constante, que representan objetos
  - Símbolos de predicado, que representan relaciones
- Todos los símbolos empiezan por mayúscula
- Cada símbolo de predicado tiene su aridad que fija su número de argumentos
- Cada modelo incluye una interpretación que dice qué objetos y relaciones se corresponden con los símbolos de constante y predicado

Ejemplo de símbolos e interpretaciones



- **Símbolos de constante**: *Richard*, *John*
- **Símbolos de predicado**: *Brother*, *OnHead*, *Person*, *King*, *Crown*
- Una **posible interpretación** (entre otras muchas) :
  - *Richard* se refiere a Ricardo Corazón de León y *John* se refiere al malvado rey Juan
  - *Brother* se refiere a la relación de hermandad; *OnHead* se refiere a la relación “sobre la cabeza”; *Person*, *King* y *Crown* se refieren a los conjuntos de objetos que son personas, reyes y coronas, respectivamente

Sintaxis y Semántica

Términos, Fórmulas Atómicas y Compuestas

Términos

- Un término es una expresión lógica que hace referencia a un objeto
  - Los símbolos de constante son términos
- Ejemplos de términos:
  - *Richard*,
  - *John*

Fórmulas atómicas

- Una fórmula atómica (también llamada átomo) es un símbolo de predicado seguido de una lista de términos que hacen de argumentos
- Una sentencia atómica es verdadera en un determinado modelo si la relación a la que se refiere el símbolo de predicado se cumple entre los objetos a los que se refieren los argumentos
- Ejemplos de átomos:
  - *Brother*(*Richard*,*John*),
  - *Person*(*Richard*)

Fórmulas compuestas

- Conectivas lógicas forman fórmulas compuestas a partir de los átomos, con la misma sintaxis y semántica que en la lógica proposicional
- Ejemplos de fórmulas compuestas:
  - $Brother(Richard, John) \wedge Brother(John, Richard)$
  - $King(Richard) \vee King(John)$
  - $\neg King(Richard) \longrightarrow King(John)$
    - (todas ellas son verdaderas en nuestro ejemplo de modelo bajo la interpretación considerada anteriormente)

Sintaxis y Semántica

Cuantificadores

Cuantificadores

- Los cuantificadores nos permiten expresar propiedades de colecciones de objetos

- El cuantificador universal $\forall$ se lee "para todo".
  - Va seguido de una o más variables en minúsculas.
  - La fórmula $\forall x\ P$ quiere decir que P es verdadero para todo objeto x
- El cuantificador existencial $\exists$ se lee "existe".
  - Va seguido de una o más variables en minúsculas.
  - La fórmula $\exists x\ P$ quiere decir que P es verdadero para al menos un objeto x

Ejemplos de uso de cuantificadores

- ***"Todos los reyes son personas"***
  - $\forall x\ King(x) \longrightarrow Person(x)$ ***(correcto)***
  - $\forall x\ King(x) \land Person(x)$ (error)
    - ***"Todos los objetos son reyes y personas"***

- ***"El rey Juan tiene una corona sobre su cabeza"***
  - $\exists x\ Crown(x) \land OnHead(x, John)$ ***(correcto)***
  - $\exists x\ Crown(x) \longrightarrow OnHead(x, John)$ (error)
    - ***"Existe un objeto que o no es una corona o está sobre la cabeza del rey Juan"***

Más acerca de los cuantificadores

- Los cuantificadores se pueden anidar.
- Si combinamos existenciales con universales, el orden es muy importante:
  - "Todo el mundo ama a alguien": $\forall x \; \exists y \; Loves(x,y)$
  - "Existe alguien que es amado por todo el mundo": $\exists y \; \forall x \; Loves(x,y)$
- Reglas de De Morgan para fórmulas cuantificadas:
  - $\forall x \; P \equiv \neg \; \exists x \; \neg P$
  - $\exists x \; P \equiv \neg \; \forall x \; \neg P$

Igualdad

- Podemos usar el símbolo de igualdad = para indicar que dos términos se refieren al mismo objeto
  - Si queremos expresar que el objeto al que se refiere *Father*(*John*) y el objeto al que se refiere *Henry* son el mismo, escribimos:

    $$Father(John)=Henry$$

  - "Ricardo tiene al menos dos hermanos":

    $$\exists x,y\ Brother(x,Richard) \wedge Brother(y,Richard) \wedge \neg(x=y)$$

Ejercicio 9.6

Ejercicio 9.6 de la tercera edición del libro. Escribe representaciones lógicas para los siguientes enunciados:

a. Los caballos, las vacas y los cerdos son mamíferos.
b. La cría de un caballo es un caballo.
c. Bluebeard es un caballo.
d. Bluebeard es un progenitor de Charlie.
e. Cría y progenitor son relaciones inversas.

Ejercicio 9.6

a. Los caballos, las vacas y los cerdos son mamíferos.
Caballo(x) ⟶**Mamimero(x)**
Vaca(x) ⟶**Mamimero(x)**
Cerdo(x) ⟶**Mamimero(x)**

b. La cría de un caballo es un caballo.
Cria(x,y) ∧ Caballlo (y) ⟶ Caballo (X)

c. Bluebeard es un caballo.
Caballo(Bluebeard)

d. Bluebeard es un progenitor de Charlie.
Progenitor(Bluebeard, Charlie)

e. Cría y progenitor son relaciones inversas.
Cria(x,y) ⟶**Progenitor(y,x)**
Progenitor(x,y) ⟶ Cria(y,x)

Combinación cuantificadores

Usa el predicado Loves, donde Loves(x,y) quiere decir "x ama a y".

a) "Hay alguien que ama a todo el mundo".

b) "Hay alguien que ama a al menos una persona".

c) "Hay alguien que ama a algún otro".

d) "Todos se aman mutuamente".

e) "Hay alguien que es amado por todos".

f) "Hay alguien a quien todos aman".

g) "Todo el mundo tiene a alguien que lo ama".

Combinación cuantificadores

a) "Hay alguien que ama a todo el mundo".

$$\exists x\ \forall y\ Loves(x,y)$$

b) "Hay alguien que ama a al menos una persona".

$$\exists x\ \exists y\ Loves(x,y)$$

c) "Hay alguien que ama a algún otro".

$$\exists x\ \exists y\ Loves(x,y) \land x \neq y$$

d) "Todos se aman mutuamente".

$$\forall x\ \forall y\ Loves(x,y)$$

e) "Hay alguien que es amado por todos".

$$\exists x\ \forall y\ Loves(y,x)$$

f) "Hay alguien a quien todos aman".

$$\exists x\ \forall y\ Loves(y,x)$$

g) "Todo el mundo tiene a alguien que lo ama".

$$\forall x\ \exists y\ Loves(y,x)$$

Sistemas Inteligentes

José A. Montenegro Montes
monte@lcc.uma.es