Pontificia Universidad JAVERIANA Bogotá

Inteligencia Artificial

Representación del Conocimiento
Lógica e Inferencia

Ing. Andrea Rueda, PhD
Ing. Enrique González, PhD
Ing. Abraham Montes, MSc
Departamento de Ingeniería de Sistemas

Agenda – Lógica e inferencia

1 – Lógica de primer orden

- Representación del conocimiento
- Inferencia hacia adelante y hacia atrás

2 – Inferencia por Resolución

- Demostración por refutación
- Forma Clausal Conjuntiva
- Unificación

3 – Proyecto 2

- Demostrador de teoremas por resolución

Objetivo de la Representación del Conocimiento

- Razonar, usando algún conocimiento, para generar una conclusión.

Ejemplo:

Considere las siguientes frases:

- Si no llovió, Harry visitó a Hagrid hoy.
- Harry visitó a Hagrid o a Dumbledore hoy, pero no a ambos.
- Harry visitó a Dumbledore hoy.

Harry no visitó a Hagrid

Es posible responder la pregunta: ¿**Llovió hoy?**

Si, hoy llovió

Lógica Proposicional

Lógica Proposicional

- Medio de representación del conocimiento.
- Lenguaje simple.
- Consiste de símbolos de proposición y conectores lógicos.
- Maneja proposiciones verdaderas, falsas o desconocidas.

Símbolos:

- Letras mayúsculas: *P*, *Q*, *W*, *Norte*
- Significado: *semántica preestablecida*
- Valoración: *Verdadero* y *Falso*

Conectores Lógicos

- Negación: $\neg$
- Conjunción (y): $\wedge$
- Disyunción (o): $\vee$
- Implicación: $\Rightarrow$ ($\supset$, $\longrightarrow$)
- Bicondicional: $\Leftrightarrow$ ($\equiv$)

Lógica Proposicional

Lógica Proposicional

- Calcular el valor de verdad de cualquier sentencia.
  - Una sentencia es una combinación de símbolos y conectores lógicos.
  - Evaluación a partir del modelo del estado actual del mundo.

Reglas – Operadores Lógicos

- $\neg P$ es verdadero si y sólo si P es falso en el modelo.
- $P \wedge Q$ es verdadero si y sólo si P y Q (ambos) son verdaderos en el modelo.
- $P \vee Q$ es verdadero si y sólo si P o Q (alguno) es verdadero en el modelo.
- $P \Rightarrow Q$ es verdadero a no ser que P es verdadero y Q es falso en el modelo.
- $P \Leftrightarrow Q$ es verdadero si y sólo si P y Q son ambos verdaderos o ambos falsos en el modelo.

Lógica Proposicional

Pontificia Universidad JAVERIANA Bogotá

Lógica Proposicional

- Calcular el valor de verdad de cualquier sentencia.
  - Una sentencia es una combinación de símbolos y conectores lógicos.
  - Evaluación a partir del modelo del estado actual del mundo.

Reglas – Tablas de Verdad

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
falso	falso	verdadero	falso	falso	verdadero	verdadero
falso	verdadero	verdadero	falso	verdadero	verdadero	falso
verdadero	falso	falso	falso	verdadero	falso	falso
verdadero	verdadero	falso	verdadero	verdadero	verdadero	verdadero

Lógica Proposicional

- Calcular el valor de verdad de cualquier sentencia.
  - Una sentencia es una combinación de símbolos y conectores lógicos.
  - Evaluación a partir del modelo del estado actual del mundo.

Reglas – Ejemplo

- P: “Está lloviendo”
- Q: “Estoy adentro”

- $\neg P$: “No está lloviendo”
- $P \wedge Q$: “Está lloviendo y estoy adentro”
- $P \vee Q$: “Está lloviendo o estoy adentro”
- $P \Rightarrow Q$: “Si está lloviendo, entonces estoy adentro”
- $P \Leftrightarrow Q$: “Está lloviendo si y sólo si estoy adentro”

Lógica de Primer Orden

Lógica de Primer Orden

- Mayor poder de representación/expresión que la lógica proposicional.
- Existencia de reglas, expresa relaciones, no sólo de hechos.
- Añade términos para representar objetos, así como cuantificadores universales y existenciales.

Símbolos

- Constantes: objetos
  Minerva, Pomona, Gryffindor
- Predicados: relaciones o funciones

  Casa(c), Persona(p)

  PerteneceA(p,c)

Cuantificadores

- Universal: “Para todo …”
  $\forall x\ \ PerteneceA(x, Gryffindor) \Rightarrow \neg PerteneceA(x, Hufflepuff)$
- Existencial: “Para algún …”, “Existe uno para el cual …”
  $\exists x\ \ Casa(x) \land PerteneceA(Minerva, x)$

Lógica de Primer Orden – Inferencia

Lógica de Primer Orden

- Mayor poder de representación/expresión que la lógica proposicional.
- Existencia de reglas, expresa relaciones, no sólo de hechos.
- Añade términos para representar objetos, así como cuantificadores universales y existenciales.

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Lógica de Primer Orden - Inferencia

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Representación de Reglas de Inferencia

$$\frac{\text{Premisa, lo que se conoce}}{\text{Conclusión, lo que se infiere}}$$

Modus Ponens

$$\frac{\alpha \Rightarrow \beta,\ \alpha}{\beta}$$

$$\frac{\text{“Si está lloviendo, entonces Harry está adentro”} \quad \text{“Está lloviendo”}}{\text{“Harry está adentro”}}$$

Lógica de Primer Orden - Inferencia

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Reglas de Simplificación

$$\frac{\text{Cláusula Inicial}}{\text{Cláusula Simplificada}}$$

Eliminación de Conjunción

$$\frac{\alpha \wedge \beta}{\alpha} \qquad \frac{\alpha \wedge \beta}{\beta}$$

$$\frac{\text{“Harry es amigo de Hermione y Ron”}}{\text{“Harry es amigo de Hermione”}}$$

$$\frac{\text{“Harry es amigo de Hermione y Ron”}}{\text{“Harry es amigo de Ron”}}$$

Lógica de Primer Orden - Inferencia

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Reglas de Simplificación

$$\frac{\text{Cláusula Inicial}}{\text{Cláusula Simplificada}}$$

Eliminación de Doble Negación

$$\frac{\neg(\neg\alpha)}{\alpha}$$

$$\frac{\text{“No es cierto que Harry no pasó el examen”}}{\text{“Harry pasó el examen”}}$$

Lógica de Primer Orden - Inferencia

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Reglas de Simplificación

$$\frac{\text{Cláusula Inicial}}{\text{Cláusula Simplificada}}$$

Eliminación de Implicación

$$\frac{\alpha \Rightarrow \beta}{\neg\alpha \lor \beta}$$

$$\frac{\text{“Si está lloviendo, entonces Harry está adentro”}}{\text{“No está lloviendo o Harry está adentro”}}$$

Lógica de Primer Orden - Inferencia

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Reglas de Simplificación

$$\frac{\text{Cláusula Inicial}}{\text{Cláusula Simplificada}}$$

Eliminación de Bicondicional

$$\frac{\boldsymbol{\alpha} \Leftrightarrow \boldsymbol{\beta}}{(\boldsymbol{\alpha} \Rightarrow \boldsymbol{\beta}) \land (\boldsymbol{\beta} \Rightarrow \boldsymbol{\alpha})}$$

“Está lloviendo si y sólo si Harry está adentro”

“Si está lloviendo, entonces Harry está adentro” y
“Si Harry está adentro, entonces está lloviendo”

Lógica de Primer Orden - Inferencia

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Reglas de Simplificación

$$\frac{\text{Cláusula Inicial}}{\text{Cláusula Simplificada}}$$

Leyes de Morgan - Disyunción

$$\frac{\neg(\alpha \lor \beta)}{\neg\alpha \land \neg\beta}$$

$$\frac{\text{“No es cierto que Harry o Ron pasaron el examen”}}{\text{“Harry no pasó el examen” y “Ron no pasó el examen”}}$$

Lógica de Primer Orden - Inferencia

Pontificia Universidad JAVERIANA Bogotá

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Reglas de Simplificación

$$\frac{\text{Cláusula Inicial}}{\text{Cláusula Simplificada}}$$

Leyes de Morgan - Conjunción

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \vee \neg\beta}$$

“No es cierto que Harry y Ron pasaron el examen”

“Harry no pasó el examen” o

“Ron no pasó el examen”

Inferencia

- Reglas para generar nueva información a partir del conocimiento existente.
- Para qué?

  Derivar una prueba, una secuencia de conclusiones que conducen al resultado deseado.

Reglas de Simplificación

$$\frac{\text{Cláusula Inicial}}{\text{Cláusula Simplificada}}$$

Propiedad Distributiva

$$\frac{(\alpha \land (\beta \lor \gamma))}{(\alpha \land \beta) \lor (\alpha \land \gamma)}$$

$$\frac{(\alpha \lor (\beta \land \gamma))}{(\alpha \lor \beta) \land (\alpha \lor \gamma)}$$

Ejemplo Inferencia

1. Marco era un hombre:
   Hombre(Marco)
2. César fue un gobernante:
   Gobernante(Cesar)
3. Todo el mundo es leal a alguien:
   $\forall x\ \exists y\ Leal(x,y)$
4. Marco intentó asesinar a Cesar:
   IntentaAsesinar(Marco,Cesar)
5. La gente sólo intenta asesinar a los gobernantes a los que no es leal.
   $\forall x\ \exists y\ Hombre(x) \wedge Gobernante(y) \wedge IntentaAsesinar(x,y) \Rightarrow \neg Leal(x,y)$

¿Era Marco leal a César?

Lógica de Primer Orden - Inferencia

Ejemplo Inferencia hacia Adelante

- Llegar a una conclusión verificando el antecedente de una regla
- El antecedente es verdadero si es soportado por las demás sentencias

1. Hombre(Marco)
2. Gobernante(Cesar)
3. ∀x ∃y Leal(x,y)
4. IntentaAsesinar(Marco,Cesar)
5. ∀x ∃y Hombre(x) ∧ Gobernante(y) ∧ IntentaAsesinar(x,y) ⇒ ¬Leal(x,y)

Marco no era leal a César

¬Leal(Marco,Cesar)

x=Marco
y=Cesar

(5) Hombre(x) ^ Gobernante(y) ^ IntentaAsesinar(x,y) → ¬Leal(x,y)

x=Marco

Hombre(Marco)
(1)

y=Cesar

Gobernante(Cesar)
(2)

x=Marco
y=Cesar

IntentaAsesinar(Marco,Cesar)
(4)

Lógica de Primer Orden - Inferencia

Pontificia Universidad JAVERIANA Bogotá

Ejemplo Inferencia Hacia Atrás

- Comprobar que el consecuente de una regla es verdadero
- Verificando que el precedente es verdadero

Marco no era leal a César

¬Leal(Marco,Cesar)

x=Marco

y=Cesar

Hombre(Marco) ^ Gobernante(Cesar) ^ IntentaAsesinar(Marco,Cesar) → ¬Leal(Marco,Cesar)

1. Hombre(Marco)
2. Gobernante(Cesar)
3. ∀x ∃y Leal(x,y)
4. IntentaAsesinar(Marco,Cesar)
5. ∀x ∃y Hombre(x) ∧ Gobernante(y) ∧ IntentaAsesinar(x,y) ⇒ ¬Leal(x,y)

Hombre(Marco) ^ Gobernante(Cesar) ^ IntentaAsesinar(Marco,Cesar)

IntentaAsesinar(Marco,Cesar)

(4)

Gobernante(Cesar)

(2)

Hombre(Marco)

(1)

Inferencia por Resolución

Demostración por Refutación

- Prueba por contradicción: para demostrar que la base de conocimiento (KB) implica una sentencia **α** dada, se demuestra que (KB **∧** ¬**α**) no puede satisfacerse.
- Al aplicar sucesivamente la regla de resolución unitaria se llega a una cláusula vacía.

Forma Normal Conjuntiva - Forma Clausal Conjuntiva

- Cada sentencia en lógica proposicional es lógicamente equivalente a una conjunción de cláusulas.

Resolución Unitaria

- Regla de resolución unitaria: siendo a_i y b literales complementarios ($a_i = \neg b$)

$$\frac{(a_1 \wedge \ldots \wedge a_{i-1} \wedge \neg b \wedge a_{i+1} \wedge \ldots \wedge a_k),\ \ b}{(a_1 \wedge \ldots \wedge a_{i-1} \wedge a_{i+1} \wedge \ldots \wedge a_k)}$$

se eliminan tanto $a_i = \neg b$ como b

$$\frac{(\textit{frío} \vee \text{calor}),\ \neg\text{calor}}{\text{frío}}$$

Inferencia por Resolución

Convertir a Forma Normal Conjuntiva

1. Eliminar $\Leftrightarrow$
   reemplazando $\boldsymbol{\alpha \Leftrightarrow \beta}$ por $(\boldsymbol{\alpha \Rightarrow \beta}) \wedge (\boldsymbol{\beta \Rightarrow \alpha})$

2. Eliminar $\Rightarrow$
   reemplazando $\boxed{\boldsymbol{\alpha \Rightarrow \beta} \quad \text{por} \quad \neg\boldsymbol{\alpha} \vee \boldsymbol{\beta}}$

3. Negaciones sólo en literales
   reemplazando
   $\neg(\neg\boldsymbol{\alpha})$ por $\boldsymbol{\alpha}$
   $\neg(\boldsymbol{\alpha} \wedge \boldsymbol{\beta})$ por $(\neg\boldsymbol{\alpha} \vee \neg\boldsymbol{\beta})$
   $\neg(\boldsymbol{\alpha} \vee \boldsymbol{\beta})$ por $(\neg\boldsymbol{\alpha} \wedge \neg\boldsymbol{\beta})$
   $\neg\forall x\, P(x)$ por $\exists x\, \neg P(x)$
   $\neg\exists x\, P(x)$ por $\forall x\, \neg P(x)$

EnCasa(Jill) ⇔ EnCasa(Novio(Jack,Jill))
EnCasa(Jill) ⇒ EnCasa(Novio(Jack,Jill)) ∧ EnCasa(Novio(Jack,Jill)) ⇒ EnCasa(Jill)

EnCasa(Jill) ⇒ EnCasa(Novio(Jack,Jill))
¬EnCasa(Jill) ∨ EnCasa(Novio(Jack,Jill))

¬(Gusta(Ana,Brocoli) ∧
¬Gusta(Ana,Brocoli) ∨ ¬Gusta(Ana,Coliflor)

¬∃x Gusta(x,Brocoli)
∀x ¬Gusta(x,Brocoli)

Inferencia por Resolución

Pontificia Universidad JAVERIANA Bogotá

Convertir a Forma Normal Conjuntiva

4. Cuantificadores con variables únicas
   reemplazando $\forall x\ P(x) \lor \forall x\ Q(x)$
   por $\forall x\ P(x) \lor \forall y\ Q(y)$

$\forall x\ (\exists y\ Animal(y) \land \neg Ama(x,y)) \lor (\exists y$
$\forall x\ (\exists y\ Animal(y) \land \neg Ama(x,y)) \lor (\exists z\ Ama(z,x))$

5. Mover cuantificadores a la izquierda
   reemplazando $\forall x\ (P(x) \lor (\forall y\ \forall z\ Q(y) \lor R(x,z)))$
   por $\forall x\ \forall y\ \forall z\ P(x) \lor Q(y) \lor R(x,z)$

$\forall x\ (\exists y\ Animal(y) \land \neg Ama(x,y)) \lor (\exists z$
$\forall x\ \exists y\ \exists z\ (Animal(y) \land \neg Ama(x,y)) \lor Ama(z,x)$

6. Eliminar cuantificadores existenciales
   reemplazando $\exists y\ P(y)$
   por $P(s1)$

$\exists y\ (Sobre(x,y) \land \neg Piramide(y))$
$Sobre(x,F(x)) \land \neg Piramide(F(x))$

$\forall x\ \exists y\ \exists z\ (Animal(y) \land \neg Ama(x,y)) \lor Ama(z,x)$
$\forall x\ (Animal(F(x)) \land \neg Ama(x,F(x))) \lor Ama(G(x),x)$

Inferencia por Resolución

Convertir a Forma Normal Conjuntiva

7. Eliminar cuantificadores universales
   reemplazando $\forall x\ P(x)$
   por $P(x)$

∀x (Animal(F(x)) ∧ ¬Ama(x,F(x))) ∨ Ama(G(x),x)
(Animal(F(x)) ∧ ¬Ama(x,F(x))) ∨ Ama(G(x),x)

8. Mover disyunciones a los literales (conjunción de disyunciones)
   reemplazando $(\boldsymbol{\alpha} \land \boldsymbol{\beta}) \lor \boldsymbol{\gamma}$
   por $(\boldsymbol{\alpha} \lor \boldsymbol{\gamma}) \land (\boldsymbol{\beta} \lor \boldsymbol{\gamma})$

(Animal(F(x)) ∧ ¬Ama(x,F(x))) ∨ Ama(G(x),x)
(Animal(F(x)) ∨ Ama(G(x),x)) ∧ (¬Ama(x,F(x)) ∨ Ama(G(x),x))

9. Separar cada parte de la conjunción como cláusulas individuales
   reemplazando $(\boldsymbol{\alpha} \lor \boldsymbol{\gamma}) \land (\boldsymbol{\beta} \lor \boldsymbol{\gamma})$
   por $(\boldsymbol{\alpha} \lor \boldsymbol{\gamma})$
   $(\boldsymbol{\beta} \lor \boldsymbol{\gamma})$

(Animal(F(x)) ∨ Ama(G(x),x)) ∧ (¬Ama(x,F(x)) ∨
Animal(F(x)) ∨ Ama(G(x),x) ¬Ama(x,F(x)) ∨ Ama(G(x),x)

Algoritmo de Unificación

- La inferencia requiere encontrar sustituciones para que dos expresiones lógicas diferentes se vean idénticas.

Ejemplo:

¿A quién conoce John?

Conoce(John,x) *En la base de conocimiento:*

Conoce(John,Jane) con {x/Jane}

Conoce(y, Bill) con {x/Bill, y/John}

Conoce(y, Madre(y)) con {y/John, x/Madre(John)}

Inferencia por Resolución

Pontificia Universidad JAVERIANA Bogotá

Algoritmo de Inferencia Prueba por Resolución

1. Negar la sentencia a probar, y añadir el resultado a la lista de axiomas (base de conocimiento).
2. Poner la lista de axiomas en Forma Normal Conjuntiva.
3. Mientras haya cláusulas por resolver:
   1) Encontrar cláusulas por resolver y generar el resultado de la resolución.
   2) Añadir resultado de la resolución a la lista de cláusulas.
   3) Si se produce la cláusula nula, detenerse y reportar que la sentencia es verdadera.
4. Detenerse y reportar que la sentencia es falsa.

Inferencia por Resolución - Ejemplo

Ejemplo:

Base de conocimiento en lenguaje natural

1. Marco es un hombre.
2. Marco es un Pompeyano.
3. Todos los Pompeyanos son Romanos.
4. César es un gobernante.
5. Todos los Romanos son o leales al César o odian al César.
6. La gente sólo intenta asesinar a los gobernantes a los que no es leal.
7. Marco intentó asesinar al César.

Pregunta: ¿Marco odia al César?

Inferencia por Resolución - Ejemplo

Ejemplo:

Base de conocimiento en lógica de primer orden

Constantes: Marco, Cesar

Predicados: Hombre(x), Pompeyano(x), Romano(x), Gobernante(x), Leal(x,y), Odia(x,y), IntentaAsesinar(x,y)

1.
2.
3.
4.
5.
6.
7.

Pregunta:

Inferencia por Resolución - Ejemplo

Pontificia Universidad JAVERIANA Bogotá

Ejemplo:

Base de conocimiento en lógica de primer orden

Constantes: Marco, Cesar

Predicados: Hombre(x), Pompeyano(x), Romano(x), Gobernante(x), Leal(x,y), Odia(x,y), IntentaAsesinar(x,y)

1. Hombre(Marco)
2. Pompeyano(Marco)
3. $\forall x$ Pompeyano(x) $\Rightarrow$ Romano(x)
4. Gobernante(Cesar)
5. $\forall x$ Romano(x) $\Rightarrow$ (Leal(x, Cesar) $\vee$ Odia(x, Cesar))
6. $\forall x\ \forall y$ (Hombre(x) $\wedge$ Gobernante(y) $\wedge$ IntentaAsesinar(x, y)) $\Rightarrow$ $\neg$Leal(x, y)
7. IntentaAsesinar(Marco, Cesar)

1. Marco es un hombre.
2. Marco es un Pompeyano.
3. Todos los Pompeyanos son Romanos.
4. César es un gobernante.
5. Todos los Romanos son o leales al César o odian al César.
6. La gente sólo intenta asesinar a los gobernantes a los que no es leal.
7. Marco intentó asesinar al César.

Pregunta: Odia(Marco, Cesar) **Negación**: 8. $\neg$Odia(Marco, Cesar)

Inferencia por Resolución - Ejemplo

Pontificia Universidad JAVERIANA Bogotá

Ejemplo:

Base de conocimiento en Forma Normal Conjuntiva

1. Hombre(Marco)
2. Pompeyano(Marco)
3. ¬Pompeyano(x3) ∨ Romano(x3)
4. Gobernante(Cesar)
5. ¬Romano(x5) ∨ Leal(x5, Cesar) ∨ Odia(x5, Cesar)
6. ¬Hombre(x6) ∨ ¬Gobernante(y6) ∨ ¬IntentaAsesinar(x6, y6) ∨ ¬Leal(x6, y6)
7. IntentaAsesinar(Marco, Cesar)

Pregunta: Odia(Marco, Cesar) **Negación**: 8. ¬Odia(Marco, Cesar)

1. Hombre(Marco)
2. Pompeyano(Marco)
3. ∀x Pompeyano(x) ⇒ Romano(x)
4. Gobernante(Cesar)
5. ∀x Romano(x) ⇒ (Leal(x, Cesar) ∨ Odia(x, Cesar))
6. ∀x ∀y (Hombre(x) ∧ Gobernante(y) ∧ IntentaAsesinar(x, y)) ⇒ ¬Leal(x, y
7. IntentaAsesinar(Marco, Cesar)

Inferencia por Resolución - Ejemplo

Odia(Marco,Cesar) ?

1. Hombre(Marco)
2. Pompeyano(Marco)
3. ¬Pompeyano(x3) ˅ Romano(x3)
4. Gobernante(Cesar)
5. ¬Romano(x5) ˅ Leal(x5,Cesar) ˅ Odia(x5,Cesar)
6. ¬Hombre(x6) ˅ ¬Gobernante(y6) ˅ ¬IntentaAsesinar(x6,y6) ˅ ¬Leal(x6,y6)
7. IntentaAsesinar(Marco,Cesar)

Inferencia por Resolución - Ejemplo

1. Hombre(Marco)
2. Pompeyano(Marco)
3. ¬Pompeyano(x3) ∨ Romano(x3)
4. Gobernante(Cesar)
5. ¬Romano(x5) ∨ Leal(x5,Cesar) ∨ Odia(x5,Cesar)
6. ¬Hombre(x6) ∨ ¬Gobernante(y6) ∨ ¬IntentaAsesinar(x6,y6) ∨ ¬Leal(x6,y6)
7. IntentaAsesinar(Marco,Cesar)

Odia(Marco,Cesar) ?

¬Odia(Marco,Cesar)

¬Romano(Marco) ∨ Leal(Marco,Cesar) ∨ Odia(Marco,Cesar)

¬Romano(Marco) ∨ Leal(Marco,Cesar)

Inferencia por Resolución - Ejemplo

1. Hombre(Marco)
2. Pompeyano(Marco)
3. ¬Pompeyano(x3) ∨ Romano(x3)
4. Gobernante(Cesar)
5. ¬Romano(x5) ∨ Leal(x5,Cesar) ∨ Odia(x5,Cesar)
6. ¬Hombre(x6) ∨ ¬Gobernante(y6) ∨ ¬IntentaAsesinar(x6,y6) ∨ ¬Leal(x6,y6)
7. IntentaAsesinar(Marco,Cesar)

Odia(Marco,Cesar) ?

¬Odia(Marco,Cesar)

¬Romano(Marco) ∨ Leal(Marco,Cesar) ∨ Odia(Marco,Cesar)

¬Pompeyano(Marco) ∨ Romano(Marco)

¬Romano(Marco) ∨ Leal(Marco,Cesar)

¬Pompeyano(Marco) ∨ Leal(Marco,Cesar)

2

Marco odia a César !!

¬Hombre(Marco) ∨ ¬Gobernante(Cesar) ∨
¬IntentaAsesinar(Marco,Cesar) ∨ ¬ Leal(Marco,Cesar)

Leal(Marco,Cesar)

1

¬Hombre(Marco) ∨ ¬Gobernante(Cesar) ∨ ¬IntentaAsesinar(Marco,Cesar)

¬Gobernante(Cesar) ∨ ¬IntentaAsesinar(Marco,Cesar)

4

7

¬IntentaAsesinar(Marco,Cesar)

Ø

TALLER Inferencia por Resolución

Pontificia Universidad JAVERIANA Bogotá

Ejercicio:

Base de conocimiento en lenguaje natural

1. Todos los que aman a todos los animales son amados por alguien.
2. Cualquiera que mate un animal es amado por nadie.
3. Jack ama a todos los animales.
4. Alguno entre Jack o Curiosidad mató al gato, que se llama Tuna.
5. Todos los gatos son animales.

Pregunta: ¿La Curiosidad mató al gato?

Proyecto 2 - Resolución

Motor de Inferencia basado en Resolución

- Implementar, usando un lenguaje procedural (C++ o Python), un programa que implemente el algoritmo de resolución por refutación visto en clase.
- En el entregable básico no se requiere manejar variables, con lo cual no hay mecanismo de unificación completo.
- Buscar un ejemplo de un teorema y usarlo como caso de validación del funcionamiento del programa.

Incluir Algoritmo de Unificación de Variables

- Ampliar el algoritmo anterior para que sea capaz de realizar la unificación de variables de tipo simbólico.
- Utilizar el ejemplo visto en clase como caso de validación del funcionamiento del programa.

Bibliografía

- Russell and Norvig. Artificial Intelligence: A Modern Approach, Third Edition. Pearson, 2016.
- P.H. Winston. Artificial Intelligence. Addison-Wesley, 1993.

Inteligencia Artificial

Representación del Conocimiento
Lógica e Inferencia

Ing. Andrea Rueda, PhD – rueda-andrea@javeriana.edu.co
Ing. Enrique González, PhD – egonzal@javeriana.edu.co
Departamento de Ingeniería de Sistemas