Agentes basados en conocimiento

El componente principal de un agente basado en conocimiento es su **base de conocimiento**, **una BC es un conjunto de sentencias**, cada una de las cuales se expresa en el lenguaje de representación del conocimiento y representa alguna afirmación acerca del mundo.

Debe haber un mecanismo para **añadir sentencias nuevas** a la base de conocimiento y uno para **preguntar qué se sabe en la base de conocimiento**. Ambas tareas requieren realizar inferencias, es decir, derivar nuevas sentencias de las antiguas.

El agente mantiene una base de conocimiento que inicialmente contiene algún conocimiento de antecedentes.

Cada vez que el programa del agente es convocado, realiza dos cosas.

Primero dice a la BC lo que ha percibido,

Segundo pregunta a la BC que acción debe ejecutar, para lo cual se debe realizar un razonamiento extensivo acerca del estado actual del mundo, de los efectos de las posibles acciones, etc.

Las sentencias se expresan de acuerdo a la sintaxis del lenguaje de representación, que especifica todas las sentencias que están bien formadas. También se debe definir la **semántica**, la cual trata del significado de las sentencias. La semántica del lenguaje **define el valor de verdad** de cada sentencia respecto a cada mundo posible.

Agentes basados en conocimiento

La **semántica** del lenguaje define el valor de verdad de cada sentencia respecto a cada mundo posible.
Los **modelos** son abstracciones matemáticas que nos permiten definir la verdad o falsedad de cada sentencia que sea relevante.
La relación de implicación lógica entre las sentencias cuya notación matemática es:

$$\alpha \models \beta$$

para significar que la sentencia α implica la sentencia β es decir **si α es verdadera β también lo debe ser**.
Para comprender la implicación y la inferencia consideremos el conjunto de todas las consecuencias de la BC como un pajar y en α como en una aguja. **La implicación** es como la aguja que se encuentra en el pajar y la **inferencia** consiste en encontrarla. Si el algoritmo de inferencia *i* puede derivar α de la BC, escribimos:

$$BC \vdash_i \alpha$$

que se lee como α se deriva de la BC mediante *i* . Un algoritmo que deriva sólo sentencias implicadas es **sólido**. Un algoritmo de inferencia es **completo** si puede derivar cualquier sentencia que está implicada.

Agentes basados en conocimiento

Si una BC es verdadera en el mundo real, entonces cualquier **sentencia α que se derive** de la BC, mediante un procedimiento de inferencia sólido también será verdadera en el mundo real.

Lógica proposicional

Sintaxis: La sintaxis de la lógica proposicional nos define las sentencias que se pueden construir. Las sentencias atómicas se componen de un único símbolo proposicional.
Cada uno de estos símbolos representa una proposición que puede ser verdadera o falsa. Utilizaremos letras mayúsculas para estos símbolos: P,Q, R, y siguientes.
Las sentencias complejas se construyen a partir de sentencias más simples mediante el uso de las conectivas lógicas, que son las siguientes:

¬ (no) Un literal puede ser una **sentencia atómica** (un literal positivo) o una sentencia atómica negativa (un literal negativo)

^ (y) Una sentencia que tenga como conectiva principal ^ se denomina **conjunción**, sus componentes son los **conjuntores**.

v (o) Una sentencia que utiliza la conectiva v es una **disyunción** de los disyuntores (componentes)

=> (implica) Es una sentencia que posee **una premisa o antecedente y una conclusión o consecuente**.

Agentes basados en conocimiento

Lógica proposicional

Semántica: La semántica en lógica proposicional debe especificar como obtener el **valor de verdad** de cualquier sentencia, dado un modelo. Este proceso se realiza de forma recursiva. Todas las sentencias se construyen a partir de las sentencias atómicas y las cinco conectivas lógicas. Hace falta definir el valor de verdad de las sentencias atómicas y como calcular el valor de verdad de las sentencias construidas con los conectivos lógicos.

Verdadero es verdadero en todos los modelos y **Falso** es falso en todos los modelos

El **valor de verdad** de cada símbolo proposicional se debe especificar directamente para cada modelo.

Para las **sentencias complejas** tenemos reglas como la siguiente:

Para toda sentencia *s* y todo modelo *m* la sentencia ¬*s* es verdadera en *m* si y sólo si *s* es falsa en *m*.

Agentes basados en conocimiento

Lógica proposicional

Tabla de verdad:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
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*

Satisfacibilidad: Una sentencia es satisfactoria si es verdadera para algún modelo. Si una **sentencia α es verdadera** en un **modelo *m*** entonces decimos que ***m* satisface α**, o que *m* es un modelo de α. La satisfacibilidad se puede averiguar enumerando los modelos posibles hasta que uno satisface la sentencia.

Agentes basados en conocimiento

Lógica proposicional Encadenamiento hacia adelante y hacia atrás

Las bases de conocimiento del mundo real contienen sólo cláusulas de tipo restringido, denominadas **Cláusulas de Horn**. **Una cláusula de Horn** es una **disyunción** de literales, de los cuales, como mucho uno es positivo, esta restricción es muy importante debido a tres razones:

- Las cláusulas de Horn con exactamente un literal positivo se denominan **cláusulas positivas**. **El literal positivo se denomina cabeza y la disyunción de literales negativos cuerpo de la cláusula**. Una **cláusula positiva** que no tiene literales negativos simplemente afirma un proposición dada que se denomina **hecho**. Las **cláusulas positivas** forman la base de la programación lógica.

- **Cada cláusula de Horn** se puede escribir como una implicación cuya premisa sea una **conjunción** de literales positivos y cuya conclusión se un único literal positivo

- La inferencia con cláusulas de Horn se puede realizar mediante algoritmos de **encadenamiento hacia adelante y de encadenamiento hacia atrás**.

- Averiguar si hay o no implicación con las cláusulas de Horn se puede realizar en un tiempo que es lineal respecto del tamaño de la base de conocimiento.

Agentes basados en conocimiento

Lógica proposicional Encadenamiento hacia adelante y hacia atrás

El encadenamiento hacia adelante es un ejemplo del concepto general de razonamiento dirigido por los datos, es decir, un razonamiento en el que el foco de atención parte de los datos conocidos. Este razonamiento se puede utilizar en un agente para derivar conclusiones a partir de percepciones recibidas, a menudo, sin una petición concreta.

El encadenamiento hacia atrás, trabaja a partir de una petición. Si se sabe que la petición ***q* es verdadera**, entonces no requiere ningún trabajo. De lo contrario el algoritmo encuentra aquellas implicaciones de la base de conocimiento de las que se concluye *q*. **Si se puede probar que todas las premisas de una de esas implicaciones son verdaderas entonces *q* es verdadero**.

Agentes basados en conocimiento

Lógica de primer orden

La lógica de primer orden es lo suficientemente expresiva para representar buena parte de nuestro conocimiento de sentido común.
El lenguaje de la lógica de primer orden está construido sobre objetos y relaciones.

Sintaxis y semántica de la lógica de primer orden
Modelos en lógica de primer orden
Los modelos de la lógica de primer orden contienen objetos. El dominio de un modelo es el conjunto de objetos que contiene, a estos objetos se les denomina elementos del dominio.
Sintaxis: Símbolos e interpretaciones
Los elementos sintácticos básicos de la lógica de primer orden son los símbolos que representan **los objetos, las relaciones y las funciones**. Por consiguiente, los símbolos se agrupan en tres tipos: **símbolos de constante**, que representan objetos, **símbolos de predicado**, que representan relaciones, y **símbolos de función** que representan funciones.

Agentes basados en conocimiento

Lógica de primer orden

La **semántica** debe relacionar las sentencias con los modelos para determinar su **valor de verdad**. Para que esto ocurra, necesitamos de una interpretación que especifique exactamente qué **objetos, relaciones y funciones** son referenciados mediante símbolos de constante, predicado y de función.

Términos: un término es una expresión lógica que se refiere a un objeto. Por lo tanto, los símbolos de constante son términos.
Sentencias atómicas: representan hechos. Una sentencia atómica está compuesta por un símbolo de predicado seguido de una lista de términos entre paréntesis

Hermano(Ricardo, Juan)

Una **sentencia atómica es verdadera** en un modelo dado y bajo una interpretación dada, si la relación referenciada por el símbolo de predicado sucede entre los objetos referenciados por los argumentos.

Agentes basados en conocimiento

Lógica de primer orden

Sustitución: Si hay alguna sustitución Θ que haga que las premisas de la implicación sean idénticas a algunas de las sentencias que ya están en la base de conocimiento, entonces podemos afirmar la conclusión de la implicación, después de aplicar Θ.
Ejemplo:
Rey(x) ^ Codicioso(x) => Malvado(x)
Rey(Juan)
Codicioso(Juan)
Se sustituye x por Juan.

Encadenamiento hacia adelante
Las cláusulas positivas de primer orden son disyunciones de literales de los cuales **sólo uno es positivo**. Una cláusula positiva es atómica o es una implicación cuyo antecedente es una conjunción de literales positivos y cuyo consecuente es un único literal positivo. Los literales de primer orden pueden contener variables, cuyo caso se asume que dichas variables están cuantificadas universalmente.

Agentes basados en conocimiento

Lógica de primer orden

Encadenamiento hacia atrás

Estos algoritmos trabajan hacia atrás **desde el objetivo**, encadenando a través de las reglas hasta encontrar los hechos conocidos que soportan la demostración.

El algoritmo se invoca con una petición y devuelve el **conjunto de todas las sustituciones** que satisfacen la petición.

La lista de objetivos se puede ver como una pila a la espera de ser procesada, si todos los objetivos se pueden satisfacer, entonces la rama actual de la demostración tiene éxito.

El algoritmo toma el **primer objetivo de la lista** y encuentra cada cláusula de la base de conocimiento cuyo literal positivo, o cabeza, se unifica con el.

Cada una de las cláusulas crea una nueva llamada recursiva en la que las premisas o cuerpo, de la cláusula se añaden a la pila de objetivos.

Los **hechos** son cláusulas con **cabeza y sin cuerpo**, así, cuando el objetivo se unifica con un hecho conocido, no se añaden más subobjetivos a la pila y el objetivo se resuelve.

Agentes basados en conocimiento

Lógica de primer orden

Prolog

Los programas en **Prolog** son conjuntos de cláusulas positivas escritas en una notación algo diferente a la estándar de la lógica de primer orden. **Prolog** utiliza las letras mayúsculas en las variables y las minúsculas en las constantes. Las cláusulas están escritas con la cabeza precediendo al cuerpo, las comas separan los literales en el cuerpo, y el punto indica el final de la sentencia. Ejemplo:

criminal(X) :- americano (X), arma(Y) , vende(X,Y,Z), hostil(Z)

La ejecución de los programas en **Prolog** se hace mediante el encadenamiento hacia atrás. El **Prolog** permite un tipo de negación denominada negación por fallo.

Un objetivo negado no P se considera demostrado si el sistema falla al probar P.

Agentes basados en conocimiento

Lógica de primer orden

Resolución

Formas normales conjuntivas en lógica de primer orden

La resolución en primer orden requiere que las sentencias estén en la forma normal conjuntiva (FNC), es decir, una conjunción de cláusulas, donde **cada cláusula es una disyunción de literales.** La sentencia en FNC será insatisfacible sólo cuando la sentencia original lo sea, así disponemos de una base para hacer demostraciones mediante contradicción con sentencias en FNC.

Procedimiento de conversión a la FNC

- Eliiminación de las implicaciones
- Anidar las negaciones
- Estándarizar las variables
- Skolenizar: eliminar los cuantificadores existenciales
- Eliminar los cuantificadores universales
- Distribuir la ^ respecto de la v

Agentes basados en conocimiento

Lógica de primer orden

Resolución

Dos cláusulas que se asume están con las variables estándarizadas, y así no comparten ninguna variable, se pueden resolver si contienen literales complementarios. Los literales en lógica de primer orden son complementarios si uno se unifica con la negación del otro.

Ejemplo:

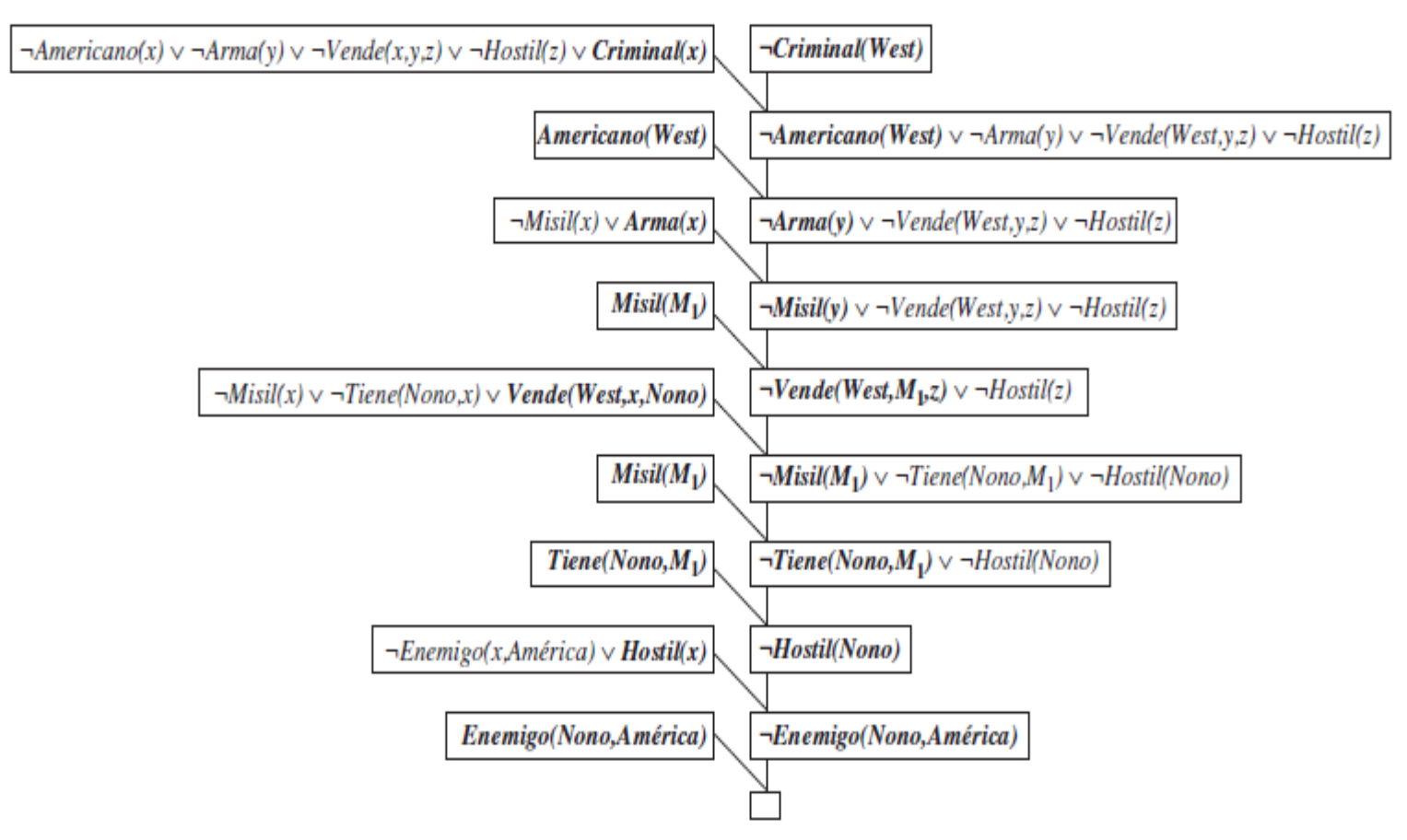
La resolución demuestra que BC |= α probando que BC ^ ¬α es insatisfacible derivando la cláusula vacía.

Las sentencias en FNC son:

$$\neg Americano(x) \vee \neg Arma(y) \vee \neg Vende(x, y, z) \vee \neg Hostil(z) \vee Criminal(x)$$
$$\neg Misil(x) \vee \neg Tiene(Nono, x) \vee Vende(West, x, Nono)$$
$$\neg Enemigo(x, América) \vee Hostil(x)$$
$$\neg Misil(x) \vee Arma(x)$$
$$Tiene(Nono, M_1). \qquad Misil(M_1)$$
$$Americano(West). \qquad Enemigo(Nono, América)$$

Agentes basados en conocimiento

Lógica de primer orden :
Resolución

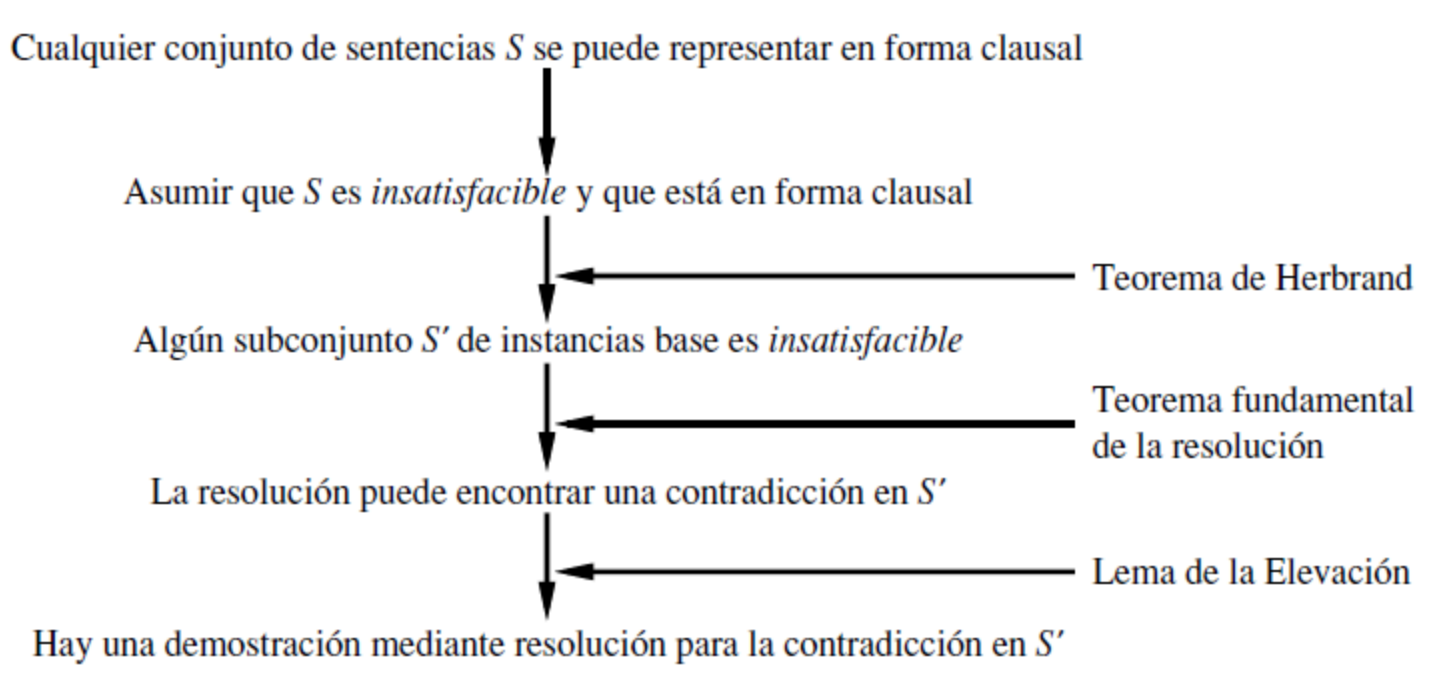


Agentes basados en conocimiento

Lógica de primer orden Completitud de la Resolución:

La resolución es un procedimiento de refutación completo, lo que significa que sí un conjunto de sentencias es insatisfacible, entonces la resolución siempre será capaz de derivar una contradicción. La resolución se puede utilizar para establecer si una sentencia se deduce de un conjunto de sentencias.

Si S es un conjunto de cláusulas insatisfacibles, entonces la aplicación de un número finito de pasos de resolución sobre S nos dará una contradicción.



Preguntas?