

Representación del conocimiento mediante lógica formal

Introducción a PROLOG

Representación del conocimiento

- Lo que nos interesa es la representación, es decir, la modelización del conocimiento
- Orientaciones:
 - **Simbólica:** la descripción del comportamiento inteligente se basa en sistemas simbólicos, más o menos formalizados
 - **Conexionista:** para describir el comportamiento inteligente se modelizan sistemas neuronales

Representación del conocimiento

- **Tratamos de encontrar la forma de modelizar que sea apropiada para el tratamiento computacional de la inferencia.**
- **Debe tener en cuenta que la inferencia se realiza sobre conocimiento incompleto o incierto (aproximado o con incertidumbre):**
 - **“No sabemos si el paciente tiene paludismo”**
 - **“Muy posiblemente el paciente tiene una infección en la faringe”**

Sistemas basados en el conocimiento

- El componente principal de un sistema basado en conocimiento (en adelante, SBC) es su **base de conocimientos**, o **BC**.
- En realidad sólo es un conjunto de representaciones de ciertos hechos acerca del mundo. A cada una de estas representaciones se le denomina **oración**. Las oraciones se representan en un lenguaje llamado **lenguaje para la representación del conocimiento**, que puede ser un lenguaje natural (como el que usamos nosotros) o un lenguaje formal (por ejemplo un lenguaje de programación).

Sistemas basados en el conocimiento

- Sobre esta base de conocimientos podría hacer varias acciones, como tratarla para llegar a una conclusión (proceso llamado inferencia) o incluso modificarla (aprendizaje) para adaptarla a otros planteamientos.
- Esquemáticamente se pueden describir las acciones que puede realizar un SBC de la siguiente forma:
 1. El SBC informa a la BC lo que percibe.
 2. Se le pregunta a la BC cuál es la acción que debe emprender. Esta puede estar decidida por un conjunto de reglas preestablecidas, o necesitar de un proceso de inferencia para llegar a la decisión.
 3. El agente realiza la acción que haya decidido de acuerdo a ese proceso de inferencia.
 4. Aprender y posiblemente modificar su BC de acuerdo a lo realizado y a si se ha conseguido el objetivo perseguido o no.

Sistemas basados en el conocimiento

- **Estas acciones se corresponden con las siguientes funciones cognitivas:**
 - **Percepción**, o cómo es la entrada de información sensorial del mundo que nos rodea.
 - **Conocimiento**, o cómo representamos esa información proveniente de los sentidos.
 - **Memoria**, o de qué forma almacenamos esa información en nuestro sistema cognitivo.
 - **Razonamiento**, o cómo tratamos esa información para realizar inferencias y llegar a conclusiones válidas.
 - **Búsqueda de soluciones**
 - **Acción**, o cual va a ser la respuesta que se va a dar en función quizá del proceso de inferencia realizado.
 - **Lenguaje natural**
 - **Procesos de consciencia**, o cómo somos conscientes de que tratamos con información, de cómo realizar una metacognición de nuestro sistema cognitivo (si eso es posible) y de cómo puedo saber que yo soy yo y no la persona con la que estoy hablando.

Sistemas basados en el conocimiento

- Aquí trataremos de los procesos de representación del conocimiento (tanto desde el punto de vista simbólico como conexionista), de los procesos de razonamiento tanto formal como informal (o con incertidumbre) y de los procesos de aprendizaje.

Formalismos de representación

- Lógicas (clásica, multivaluadas, modales y difusa)
- Reglas de producción (con incertidumbre - MYCIN)
- Redes semánticas y causales (bayesianas)
- Marcos y guiones
- Redes neuronales y sistemas neuroborrosos.

Construcción de una base de conocimientos mediante la lógica formal

- **En principio, un lenguaje de representación del conocimiento consta de dos aspectos básicos:**
 - **Sintáxis**, que explica cómo es la forma en que se construyen las oraciones como combinación de estructuras más sencillas (por ejemplo las palabras en un lenguaje natural). Así podemos decir que la oración *Un bueno será jefe justo* está sintácticamente mal construida, debido a los signos o elementos de esta oración están mal ordenados.
 - **Semántica**, que explica la correspondencia entre los elementos o signos que componen un lenguaje y los objetos que representan. Por ejemplo podemos decir que la palabra juez designa a una persona que imparte justicia.

Construcción de una base de conocimientos mediante la lógica formal

- Podríamos hacer una analogía entre forma y contenido respectivamente.
- Y a partir de aquí podemos definir el concepto de **inferencia** como el proceso que lleva de unas premisas a una conclusión, y que en principio puede ser válida o no.
- Pero, ¿qué entendemos por validez, o por una oración *válida*? Se dice que una oración es válida si y sólo si es verdadera en todos y cada uno de los casos posibles, independientemente del objeto que representen.

Construcción de una base de conocimientos mediante la lógica formal

- Así también se suele hablar de **validez formal** o de que un razonamiento es **formalmente válido** cuando dadas unas premisa **necesariamente** se obtiene una conclusión concreta. A éstas oraciones que son necesariamente verdaderas se les conoce con el nombre de **tautologías**
 - Por ejemplo, si decimos que *Todos los perros ladran* y que *Pipo es un perro*, de ésto podré concluir que *Pipo ladra*.

Tipos de razonamiento

- **A**u vez, un razonamiento puede ser **deductivo**, si la información contenida en la conclusión no aporta ningún conocimiento que no se pueda deducir de forma inmediata de las premisas (como es el caso anterior) o **inductivo** cuando de la información contenida en las premisas no se infiere de forma inmediata la conclusión.
- Por ejemplo de *Mercurio gira en una órbita elíptica, Venus gira en una órbita elíptica, la Tierra gira en una órbita elíptica* podemos inferir que *Plutón gira en una órbita elíptica*, pero eso no es necesariamente verdadero. A lo sumo será verdadero con una cierta probabilidad o **plausibilidad**, y así llamaremos a estos razonamientos **razonamientos probabilísticos** o **plausibles**. Éstos razonamientos son los que se usan en ciencia y los que aportan de verdad conocimiento.

Tipos de razonamiento

- Por último, llamaremos **modelo** a un mundo en el que una oración es verdadera bajo una determinada interpretación. Y así, habrá oraciones que serán verdaderas en un determinado modelo y otras que no.
- Por lo tanto disponemos de un lenguaje formal, que será independiente del modelo en el que se sitúe, y que nos permitirá representar todos los elementos u objetos de un modelo (o mundo) mediante **signos** de forma no ambigua, un conjunto de reglas (**reglas de formación**) para manipular esos signos y efectuar inferencias para llegar a conclusiones y un conjunto de **reglas de transformación** que permitirá transformar una combinación bien construida en otra igualmente bien construida. Estos elementos constituyen lo que se denomina **lógica**.

Lógica

- Podemos distinguir dos tipos dentro de lo que se llama **lógica clásica** (aquella que la conclusión, si es verdadera lo es con certeza, es decir, sin ningún grado de probabilidad, mientras que hay otros tipos de lógicas no clásicas, como puede ser la lógica difusa -o **Fuzzy Logic** en el que una conclusión sólo es verdadera con un cierto grado de certeza)
- **Lógica propositiva** o **lógica de enunciados** cuando los signos que manejamos son proposiciones completas (por ejemplo, *a* podría significar *el perro de Juan es negro*, y usaremos símbolos llamados **conectores** para hacer oraciones más complejas, como por ejemplo *el perro de Juan es negro y no lleva correa*).
- **Lógica de predicados de primer orden**, en el que manejamos objetos y predicados sobre objetos, así como *propiedades* (hermano de, padre de), *cuantificadores* (algunos, todos, ningún) y *conectores*.

Lógica

- **Lógica de predicados de orden superior** en el que podemos hacer inferencias sobre la lógica, es decir, tener un conocimiento que podríamos llamar metalógico.
- **Lógicas multivaluadas**, en las que tratamos con varios valores de verdad, como por ejemplo las trivaluadas de Kleen y de Lukasiewicz donde se introduce el valor $1/2$ para las proposiciones cuyo valor de verdad es indeterminado (por ejemplo las proposiciones futuras).
- **Éstas serán las herramientas de que dispondremos para representar el conocimiento. Así que vamos a ver cómo construir una base de conocimientos en base al lenguaje utilizado para representar los conceptos. En principio elegiremos la lógica de predicados como un buen lenguaje para hacerlo.**

PROLOG

- Requisitos
 - Un editor de texto: Emacs
 - Un entorno de desarrollo Prolog: CIAO Prolog, SWI Prolog o SICTUS
- Un entorno de desarrollo Prolog se compone de:
 - Un intérprete.
 - Un shell (depurador,...)

Elementos de una base de conocimientos

- **Predicados:** atributos de los elementos
- **Elementos:** proposiciones.
 - **Constantes:** se escriben en minúsculas, por ejemplo *pedro*, *gripe*.
 - **Variables lógicas:** Las variables en Prolog no son variables en el sentido habitual, por eso las llamamos variables lógicas. Se escriben como una secuencia de caracteres alfabéticos comenzando siempre por mayúscula o subrayado. Por ejemplo *Persona*, *X*,
 - **La variable anónima:** todas ellas se representan mediante el símbolo de subrayado .
- **Operadores.** El operador **Y** se representa por ,

Elementos de una base de conocimientos

- **Hechos:** expresan relaciones (predicados) entre elementos (proposiciones). Deben acabar en punto.
 - Por ejemplo *es(juan,alto)*.
- **Reglas:** condiciones **si-entonces**. Se escribe primero el consecuente y luego el/los antecedentes.
 - Por ejemplo *es(X, español):-nacio(X, madrid)*.

Proceso de razonamiento

- **Base de conocimiento: hechos y reglas.**
- **Objetivos**
- **Unificación**

El shell de Prolog

- ?- El símbolo ?- nos indica la zona donde podemos escribir los objetivos a ejecutar.
- Mi primer objetivo
Swi-Prolog 1.4 #0: Sat Nov 27 19:27:11 1999
?- t(X,3) = t(4,Z).
X = 4,
Z = 3
yes
- Observese que los objetivos acaban en un punto (.), si pulsamos intro antes de escribir el punto ocurre un salto de línea, pero nada más. Cuando escribimos el punto y pulsamos INTRO es cuando se ejecuta el objetivo.

El shell de Prolog

- A continuación, el shell nos dice si el objetivo tiene éxito o no, y cuales son las ligaduras de las variables. Después aparece un signo de interrogación (?). En este momento es cuando le podemos pedir que nos muestre otra solución tecleando un punto y coma (;) y pulsando INTRO:

```
Swi-Prolog 1.4 #0: Sat Nov 27 19:27:11 1999  
?- t(X,3) = t(4,Z).
```

```
X = 4,  
Z = 3 ? ;
```

```
no  
?-
```

El shell de Prolog

- Como no hay más soluciones en nuestro ejemplo, el shell dice "no" y nos permite escribir otro objetivo.
- Compilando y cargando código

```
?- consult('c:/prog1.pl').  
yes  
?-
```
- Obsérvese que el nombre del fichero fuente (y su ruta, si es necesario) se escribe en un término entre comillas simples. Esta es la forma habitual de escribir nombres de fichero.

Eficiencia

- Ordenación de cláusulas:
 - 1º las más específicas
 - 2º las más generales (con recursividad)

Ejemplo:

`antecesor(X,Y):-padre_de(X,Y).`

`antecesor(X,Y):-padre_de(X,Y),antecesor(Z,Y).`

- Ordenación de términos dentro de una cláusula:
 - 1º los términos más específicos
 - 2º los términos más generales (recursivos)

Ejemplo

`antecesor(X,Y):-padre_de(X,Y),antecesor(Z,Y).`

Operadores

- Desigualdad `\==` Comprueba si dos términos son distintos
- Desigualdad aritmética `=\=` Verifica la desigualdad de dos expresiones
- Operadores lógicos y aritméticos: `<` `>` `=<` `>=`
 - Aritméticos: `+` `-` `*` `/` `abs` `^(potencia)` `max` `min` ...
- Asignación `[is]`. Evalúa la segunda expresión e intenta asignar el valor obtenido a la variable. ¡No es conmutativo!
 - Ejemplo: `Incremento(X,Y):-Y is X+1`
- Negación: `not millonario(X)`, o `not(X==0)`
- Agregar hechos: `assert(padre_de(luis,pepe))`.
 - **NOTA:** deben ser procedimientos dinámicos, no estáticos. Para convertir un término en dinámico debo agregar a la BC el siguiente procedimiento:
`:-dynamic padre_de/2.`
- **Eliminar hechos:** `retract(padre_de(luis,pepe))`.

Listas

- $[a,b,c]$ Lista con tres elementos
- $[\]$ Lista vacía
- $[a|L]$ Lista con el elemento a en la cabecera y el resto en la variable L (cola)
- $[a,b|L]$
- $[X|L]$ Lista con el primer elemento instanciado en la variable X y el resto en la variable L (cola)
- $[X,Y|L]$

Predicados para el manejo de listas

- **Nº de elementos de una lista:** longitud (Lista,N): longitud([],0).
- **Pertenencia a una lista:** pertenece(X,Lista).
- **Concatenar dos listas:** concatenar(L1,L2,L3).
- **Eliminar un elemento X de una lista L, y como consecuencia se obtiene la nueva lista L2:**
elimina(X,L1,L2).
- **Invertir los elementos de una lista:** invertir(L1,L2).

Estructuras de datos en Prolog

- Ejemplo:

reino(juan,portugal).

reino(juan,suecia).

años_de_reinado(reino(duarte,portugal),1433,1438).

años_de_reinado(reino(juan,portugal),1481,1495).

años_de_reinado(reino(duarte,suecia),1568,1592).

?-años_de_reinado(reino(juan,portugal),X,Y).

X=1481.

Y=1495.

?- años_de_reinado(reino(juan,Pais),X,Y).

P = portugal

X = 1481

Y = 1495

Estructuras de datos en Prolog

- ¿Cómo podría saber si un rey ha estado reinando en algún país durante un cierto año?

reino(juan,portugal).

reino(juan,suecia).

años_de_reinado(reino(duarte,portugal),1433,1438).

años_de_reinado(reino(juan,portugal),1481,1495).

años_de_reinado(reino(duarte,suecia),1568,1592).

en_el_trono(X,P,Año):-años_de_reinado(reino(X,P),A,B),Año>=A,Año=<B.

?- en_el_trono(juan,P,1490).

P = portugal