

Agentes Inteligentes

Javier García

Departamento de Electrónica y Computación
Universidad de Santiago de Compostela

September 15, 2021

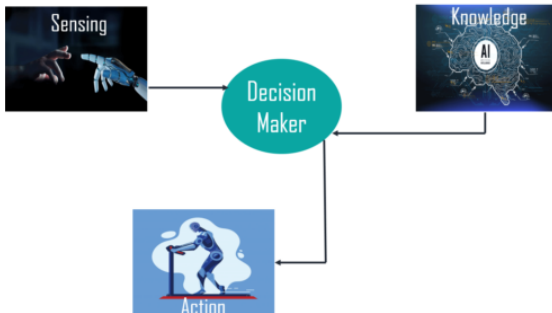
Part III

Representación del conocimiento

- 1 Introducción
- 2 Formas de representación
 - Lógica proposicional
 - Lógica de predicados
 - Otras lógicas
 - Redes semánticas
 - Objeto-Atributo-Valor
 - Marcos
 - Programación orientada a objetos
 - Espacio de búsqueda

Introducción (I)

- Para que el ordenador pueda resolver problemas, hace falta decirle qué tiene que resolver en algún lenguaje (al igual que a nosotros)
- Necesidad de representacion de las entradas del agente
- ¿Qué representar?
 - Hechos
 - Acciones
 - ...



- Existen muchas formas de suministrar esa información
 - Lógica proposicional
 - Lógica de predicados
 - Lógicas de segundo orden, difusa
 - Redes semánticas
 - Objeto-Atributo-Valor
 - Marcos
 - Programación orientada a objetos
 - Espacio de búsqueda

Lógica proposicional (booleana o de orden cero) (I)

- Elementos de representación
 - Propositiones o *variables proposicionales* que pueden ser \top o \perp
 - Símbolos lógicos o conectivas: $\wedge, \vee, \rightarrow, \neg$
- Inferencia: deducciones con reglas, hechos y *Modus-Ponens*:

$$P \rightarrow Q$$

“Si P implica Q ; y P es verdad; entonces Q también es verdad”

- Ejemplos: $\text{llueve}, (\neg \text{nieva} \wedge \text{llueve}) \vee \text{hay-hielo}$
- Ventajas: representación de tipo general, y decidible (en tiempo finito es capaz de decidir si una proposición es deducible de la información disponible o no)
- Problema: representación limitada

Lógica proposicional (booleana o de orden cero) (II)

- Una fábrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasiones

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

Lógica proposicional (booleana o de orden cero) (II)

- Una fábrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasiones

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

$$R1 : s_3 \vee f_2 \rightarrow a_1$$

- Si saltan los detectores de fuego 1 y 4, se debe producir la alarma 2

Lógica proposicional (booleana o de orden cero) (II)

- Una fábrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasiones

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

$$R1 : s_3 \vee f_2 \rightarrow a_1$$

- Si saltan los detectores de fuego 1 y 4, se debe producir la alarma 2

$$R2 : s_1 \wedge s_4 \rightarrow a_2$$

- Si salta la alarma 1, y el detector de fuego 2 o el de fugas 1, se debe producir la alarma 3

Lógica proposicional (booleana o de orden cero) (II)

- Una fábrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasiones

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

$$R1 : s_3 \vee f_2 \rightarrow a_1$$

- Si saltan los detectores de fuego 1 y 4, se debe producir la alarma 2

$$R2 : s_1 \wedge s_4 \rightarrow a_2$$

- Si salta la alarma 1, y el detector de fuego 2 o el de fugas 1, se debe producir la alarma 3

$$R3 : a_1 \wedge (s_2 \vee f_1) \rightarrow a_3$$

Lógica proposicional (booleana o de orden cero) (III)

- Inferencia: deducción
- Han saltado el detector de fuego 2 y el de fugas 2, ¿qué alarmas saltarían?

① s_2

② f_2

Lógica proposicional (booleana o de orden cero) (III)

- Inferencia: deducción
- Han saltado el detector de fuego 2 y el de fugas 2, ¿qué alarmas saltarían?
 - 1 s_2
 - 2 f_2
 - 3 Con f_2 verdadero en $R1$, se activa a_1

Lógica proposicional (booleana o de orden cero) (III)

- Inferencia: deducción
- Han saltado el detector de fuego 2 y el de fugas 2, ¿qué alarmas saltarían?
 - 1 s_2
 - 2 f_2
 - 3 Con f_2 verdadero en $R1$, se activa a_1
 - 4 Con s_2 y a_1 verdadero en $R3$, se activa a_3

Lógica proposicional (booleana o de orden cero) (IV)

- Dificultadas de representación
- Una empresa tiene 10 empleados. Los empleados pueden trabajar en tres tipos de puestos: director, jefe o administrativo

Empleado1TrabajaDeDirector, Empleado2TrabajaDeJefe,...

- Si es director gana 60000 euros brutos al año, si es jefe 30000 y, si no, 20000. Además, si tiene más de dos hijos, gana 10000 euros más al año

Empleado1Tiene1Hijo, Empleado2Tiene3Hijos,...

$\text{Empleado1Tiene1Hijo} \wedge \text{Empleado1TrabajaDeDirector} \rightarrow$
 $\text{Empleado1Gana60000}$

$\text{Empleado2Tiene3Hijos} \wedge \text{Empleado2TrabajaDeJefe} \rightarrow$
 $\text{Empleado2Gana40000}$

Lógica de predicados (de primer orden) (I)

- Elementos de representación
 - Constantes, variables
 - Predicados
 - $\text{trabaja_como}(X,Y)$
 - $\text{tiene_hijos}(X,Y)$
 - Fórmulas atómicas: predicados definidos sobre términos
 - $\text{trabaja_como}(\text{empleado1,director})$
 - $\text{tiene_hijos}(\text{empleado1,1})$
 - Formulas bien formadas: fórmulas atómicas unidas por conectividad ($\wedge, \vee, \rightarrow, \neg$) y cuantificadores (\forall, \exists)
 - $\forall X, Y \text{ trabaja_como}(X, \text{director}) \wedge \text{tiene_hijos}(X, Y) \wedge Y \leq 2 \rightarrow \text{gana}(X, 60000)$
 - $\forall X, Y \text{ trabaja_como}(X, \text{director}) \wedge \text{tiene_hijos}(X, Y) \wedge Y > 2 \rightarrow \text{gana}(X, 70000)$
- Representación de tipo general más rica que la proposicional
- Existe un lenguaje de programación que permite crear y ejecutar programas en lógica de predicados: PROLOG

“Una universidad imparte un conjunto de titulaciones en un conjunto de centros. Cada titulación tiene un plan de estudios formado por un conjunto de asignaturas troncales, obligatorias, optativas y de libre elección. Cada asignatura se imparte en un curso y cuatrimestre determinados y tiene un determinado número de créditos. Todos los alumnos de nuevo ingreso deben matricularse de todas las asignaturas del primer curso. En primero de Ingeniería Técnica en Informática de Gestión de la Universidad Carlos III de Madrid se imparte en primero las asignaturas de Matemáticas I,...Ana Morales Pérez acaba de matricularse en primero de la titulación”

Lógica de predicados (de primer orden) (III)

- Una universidad imparte un conjunto de titulaciones en un conjunto de centros y campus
 - `imparte(Universidad,Titulación,Centro,Campus)`
 - `imparte(uc3m,itig,eps,leganés)`
 - `imparte(uc3m,itig,eps,colmenarejo)`
 - `imparte(uc3m,ii,eps,leganés)`
- La representación no es única
 - `imparte_titulación(Universidad,Titulación)`
 - `imparte_titulación(uc3m,itig)`
 - `imparte_titulación(uc3m,ii)`
 - `imparte_centro(Titulación,Centro)`
 - `imparte_centro(itig,eps)`
 - `centro_en_campus(Centro,Campus)`
 - `centro_en_campus(eps,leganés)`

Lógica de predicados (de primer orden) (IV)

- Cada titulación tiene un plan de estudios formado por un conjunto de asignaturas troncales, obligatorias, optativas y de libre elección
 - `asignatura_en_plan(Asignatura,Titulación)`
 - `asignatura_en_plan(ia-itig,itig)`
 - `tipo_asignatura(Asignatura,Tipo)`
 - `tipo_asignatura(ia-itig,obligatoria)`
 - `tipo_asignatura(ia-ii,troncal)`
- Cada asignatura se imparte en un curso y cuatrimestre determinados y tiene un determinado número de créditos
 - `curso_asignatura(Asignatura,Curso)`
 - `cuatrimestre_asignatura(Asignatura,Cuatrimestre)`
 - `creditos_asignatura(Asignatura,Creditos)`
 - o `asignatura(Asignatura,Curso,Cuatrimestre,Creditos)`
- Los alumnos se matriculan en asignaturas y pueden ser de nuevo ingreso
 - `matricula_en(Alumno,Asignatura)`
 - `nuevo_ingreso(Alumno)`

- Los alumnos de nuevo ingreso, deben matricularse de todas las asignaturas del primer curso

$R1 : \forall X, Y \text{ nuevo_ingreso}(X) \wedge \text{curso_asignatura}(Y, 1) \rightarrow \text{matriculado_en}(X, Y)$

Lógica de predicados (de primer orden) (VI)

- En primero de Ingeniería Técnica en Informática de Gestión de la Universidad Carlos III de Madrid se imparte en primero las asignaturas de Matemáticas I,...
 - 1. `curso_asignatura(matemáticasI,1)`
 - 2. `curso_asignatura(física,1)`
- Ana Morales Pérez es una alumna de nuevo ingreso
 - 3. `nuevo_ingreso(anaMorales)`
- Si $X=anaMorales$, y $Y=matemáticasI$, (unificación) por *Modus-Ponens*, a partir de $R1$, y de 1 y 3, se puede deducir que `matriculado_en(anaMorales,matemáticasI)`
- Si $X=anaMorales$, y $Y=física$, (unificación) por *Modus-Ponens*, a partir de $R1$, y de 2 y 3, se puede deducir que `matriculado_en(anaMorales,física)`

Lógica de predicados (de primer orden) (VII)

- Deducción hacia atrás
- ¿Y si se desea conocer en qué asignaturas se debe matricular Ana?
- Pregunta: `matriculado_en(anaMorales,Y)`
- Se busca una implicación lógica donde aparezca `matriculado_en` en la parte derecha (puede haber más de una)
- Si se puede unificar, se intenta deducir los literales que aparezcan en la parte izquierda de la implicación
- Si existe alguna asignación de valor a las variables de la parte izquierda que permita deducir como ciertas las condiciones, se podrá deducir la pregunta de diferentes formas (diferentes valores de Y: matemáticasl, física,...)

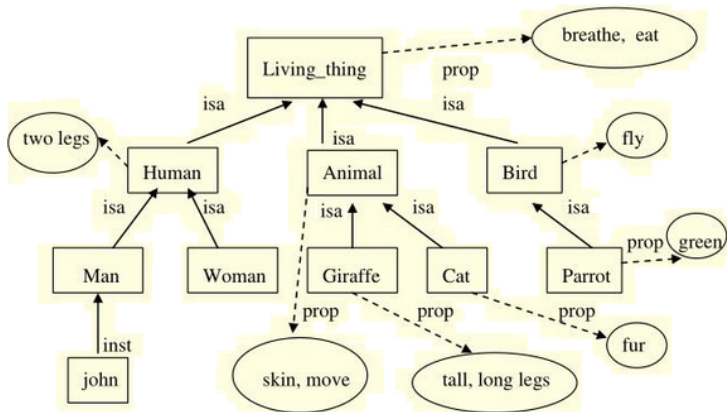
- Lógicas de segundo orden (o de orden superior)
 - Equivalente a decir que los predicados pueden tomar otros predicados como argumentos
- Lógica difusa
 - Grados de pertenencia: permite analizar información del mundo real en una escala entre lo falso y lo verdadero

Redes semánticas (I)

- Grafos dirigidos en los que los nodos representan conceptos y los arcos relaciones binarias entre ellos
- Un arco puede representar:
 - **isa**: subclase de una entidad. De una categoría a otra categoría (e.g., el hospital infantil es una subclase de un hospital)
 - **inst**: instancia particular de una clase. De un individuo a una categoría (e.g., España es una instancia de País)
 - **prop**: propiedad o característica (“ladrar” es una propiedad de los perros)
- Problemas para representar la cuantificación, la negación, la implicación (reglas), y la disyunción

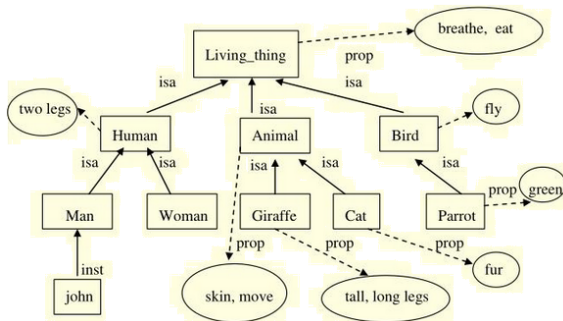
“Todos los humanos, animales y pájaros son seres vivos que comen y respiran. Todos los pájaros pueden volar. Todos los hombres y mujeres son humanos que tienen dos piernas. El gato es un animal que tiene pelaje. Todos los animales tienen piel y se pueden mover. La girafa es un animal alto con largas patas. El loro es un pájaro y es de color verde”

Redes semánticas (III)



Redes semánticas (IV)

- La idea central de las redes semánticas es la **herencia**
 - *Human* hereda las propiedades de *breathe* y *eat* de *Living_thing*
 - *Man* herada de *Human* *two legs* y *breathe* y *eat* de *Living_thing*
- Inferencia: el sistema puede inferir que los hombres comen y respiran a partir del hecho de que son seres vivos, y que todos los seres vivos comen y respiran



Objeto-Atributo-Valor

- Lista de ternas (objeto, atributo, valor)
 - (peón, está_en, casilla_1_0)
 - (casilla_1_0, X, 1)
 - (casilla_1_0, Y, 0)
 - ...
- Equivalente a registros (lenguajes de programación) o a predicados binarios (lógica de predicados)
- Otra forma de representación...

objeto	atributo	valor
tiburón	peligroso	verdadero
tiburón	locomoción	nada
tiburón	categoría	peces
canario	categoría	pájaros
canario	peligroso	falso

Marcos (I)

- Conceptos: clases, marcos (frames),...
- Subconceptos: subclasses
- Instancias: objetos
- Atributos: slots, campos, características...
- Herencia: simple o múltiple
- Facetas: atributos de atributos (valor, tipo de valores, comentario, cardinalidad,...)
- Relaciones: es-un, parte-de, empleado-de,...
- Métodos: funciones asociadas a los marcos (métodos de las clases)
- Demonios: funciones que vigilan las operaciones sobre los atributos

“Una universidad imparte un conjunto de titulaciones en un conjunto de centros. Cada titulación tiene un plan de estudios formado por un conjunto de asignaturas troncales, obligatorias, optativas y de libre elección. Cada asignatura se imparte en un curso y cuatrimestre determinados y tiene un determinado número de créditos. Todos los alumnos de nuevo ingreso deben matricularse de todas las asignaturas del primer curso. En primero de Ingeniería Técnica en Informática de Gestión de la Universidad Carlos III de Madrid se imparte en primero las asignaturas de Matemáticas I,...Ana Morales Pérez acaba de matricularse en primero de la titulación”

*“Una **universidad** imparte un conjunto de **titulaciones** en un conjunto de **centros**. Cada titulación tiene un **plan de estudios** formado por un conjunto de **asignaturas troncales, obligatorias, optativas y de libre elección**. Cada asignatura se imparte en un **curso** y **cuatrimestre** determinados y tiene un determinado **número de créditos**. Todos los **alumnos** de nuevo ingreso deben matricularse de todas las asignaturas del **primer curso**. En primero de **Ingeniería Técnica en Informática de Gestión** de la **Universidad Carlos III de Madrid** se imparte en primero las asignaturas de **Matemáticas I**,...**Ana Morales Pérez** acaba de matricularse en primero de la titulación”*

- Clases: universidad, titulación, centro, plan_de_estudios, asignatura, curso, alumno
- Subclases: troncales, obligatorias, libre_elección
- Instancias: Ingeniería Técnica en Informática de Gestión, MatemáticasI, Ana_Morales_Perez
- Atributos: nombre, titulaciones, centros, numero_alumnos, tipo (universidad), tipo, curso, cuatrimestre, número_créditos (asignatura)

Asignatura	
es_un:	
Atributo	Posibles valores/Valor
nombre	cadena de caracteres
titulación	instancia de Titulación
curso	número
cuatrimestre	número
alumnos	lista de Alumnos
número_alumnos	número

Asignatura_troncal	
es_un: Asignatura	
Atributo	Posibles valores/Valor
tipo	troncal
número_créditos	número

Titulación	
es_un:	
Atributo	Posibles valores/Valor
nombre	cadena de caracteres
universidad	instancia de Universidad
centros	lista de Centros
número_alumnos	número

Ejemplo de instancia

IngTecInfGestion	
es_un: Titulación	
Atributo	Posibles valores/Valor
nombre	Ingeniería Técnica en Informática de Gestión #UC3M { #Leganés, #Colmenarejo} 800
universidad	
centros	
número_alumnos	

Facetas

- Atributo Universidad de clase Titulación en instancia IngTecInfGestion
 - valor: puntero a instancia UC3M (de Universidad)
 - tipo de valores: instancia de clase Universidad
 - comentario: "Se refiere a la titulación de Ingeniería Técnica en Informática de Gestión de la Universidad Carlos III de Madrid"
 - cardinalidad (máxima-mínima): 1
 - demonio: -

Métodos

```
matricular.asignatura (self,alumno)
    if not(member(alumno,get(self,alumnos))) then
        add(self,alumnos,alumno)
        set(self,número_alumnos,get(self,número_alumnos)+1)
```

Demonios

```
if-added.número_alumnos.Titulación (self)
    # Cuando se da valor al atributo número_alumnos en Titulación
    número=0
    For asignatura in get(self,Titulaciones)
        número=número+get(asignatura,número_alumnos)
```

Programación orientada a objetos

- Java, C++, Python,...
- Definición de clases, subclasses (herencia), instancias (objetos)
- Muchos de los mismos principios que las representaciones anteriores...

```
public class Animal{  
    public Animal ();  
    Eye eyes;  
    Leg legs;  
    Head head;  
    Tail tail;  
}
```

```
Animal an_animal = new Animal();
```

```
class Dog extends Animal{  
    public void bark();  
}
```

Espacio de búsqueda (I)

- Muchos problemas en IA se pueden representar como espacios de búsqueda
- Un espacio de búsqueda es una representación del conjunto de opciones posibles (estados) en un problema dado, una o más de las cuales son la solución del problema
- Espacio de búsqueda = Espacio de estados
- No una forma de representación de conocimiento *per se*, representa la estructura del problema en términos de las alternativas disponibles en cada posible estado
- Representación de estados y operadores
 - Búsqueda clásica: Objeto-Atributo-Valor, Programación orientada a objetos,...
 - Planificación Automática: Lógica de predicados
- Generalmente se busca encontrar el camino entre un estado inicial y un estado final (meta)

Espacio de búsqueda (II)

