

# Tema 1: Agentes inteligentes: Representación y razonamiento

José A. Alonso Jiménez  
Miguel A. Gutiérrez Naranjo  
Francisco J. Martín Mateos

Dpto. de Ciencias de la Computación e Inteligencia Artificial

UNIVERSIDAD DE SEVILLA

# Contenido

- **Agentes inteligentes: Representación y razonamiento**
  - ¿Qué es la inteligencia computacional?
  - Agentes en el mundo
  - Representación y razonamiento
  - Ejemplos de agentes
  - Sistemas de representación y razonamiento
  - Casos de estudios

## ¿Qué es la inteligencia computacional?

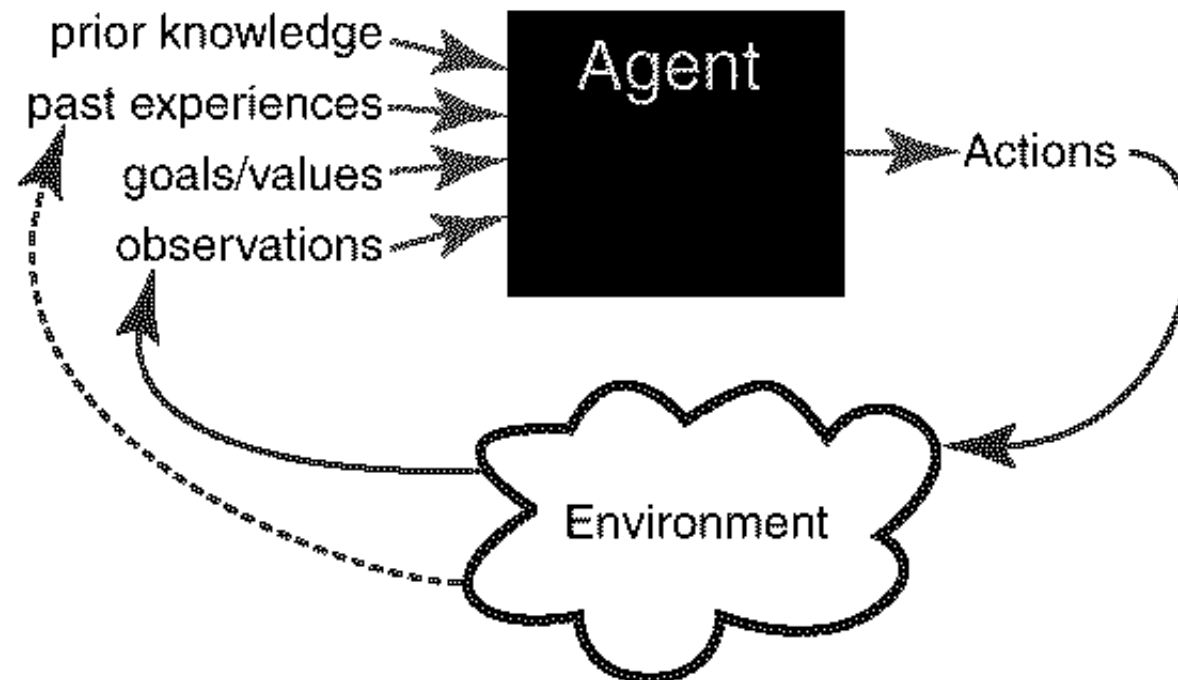
- La *inteligencia computacional* es el estudio de los agentes inteligentes.
- Un *agente* es cualquiera que actúa en un entorno.
- Un *agente inteligente* es un agente que actúa inteligentemente:
  - sus acciones son adecuadas a sus objetivos y valores,
  - es flexible a los cambios de entornos y objetivos,
  - aprende de la experiencia,
  - toma decisiones con conocimiento incompleto.

# ¿Inteligencia artificial o computacional?

- *Objetivos* de la Inteligencia Computacional:
  - *Objetivo científico*: comprender los principios del comportamiento inteligente
  - *Objetivo tecnológico*: especificar métodos para diseñar sistemas inteligentes
- Analogía entre máquinas que vuelan y máquinas que piensan
- *Hipótesis del sistema de símbolos*:
  - El razonamiento es manipulación de símbolos
  - Razonamiento = Computación
- *Tesis de Turing*:
  - Cualquier manipulación de símbolos puede realizarse mediante una máquina de Turing

# Agentes en el mundo

- Mundo = Agente + Entorno (Poole-98 p. 8)



# Agentes en el mundo

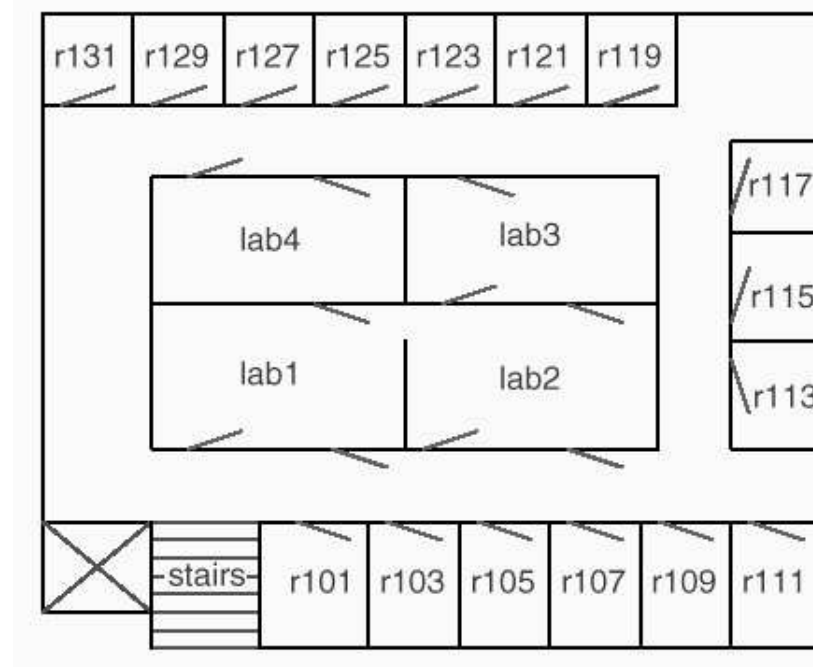
- Entradas del agente
  - *Conocimiento previo* del mundo
  - *Experiencias anteriores* de las que puede aprender
  - *Objetivos* a conseguir y *valores* sobre lo importante
  - *Observaciones* sobre su entorno y sobre sí mismo
- Salidas del agente
  - Acciones

# Representación y razonamiento

- Necesidad de representación de las entradas del agente: *conocimiento*
- Problema  $\implies$  Representación  $\implies$  Razonamiento
- Elementos de un *Sistema de Representación y Razonamiento* (SRR):
  - *Sintaxis*: Lenguaje de comunicación con la computadora
  - *Semántica*: Manera de asignar significado al lenguaje
  - *Cálculo*: Procedimientos para obtener respuestas
- Ejemplos de SRR:
  - Lenguajes de bajo nivel: Fortran, C, Lisp, ...
  - Lenguaje natural

# El robot repartidor

- El mundo del robot repartidor (Poole-98 p. 14)





# El robot repartidor

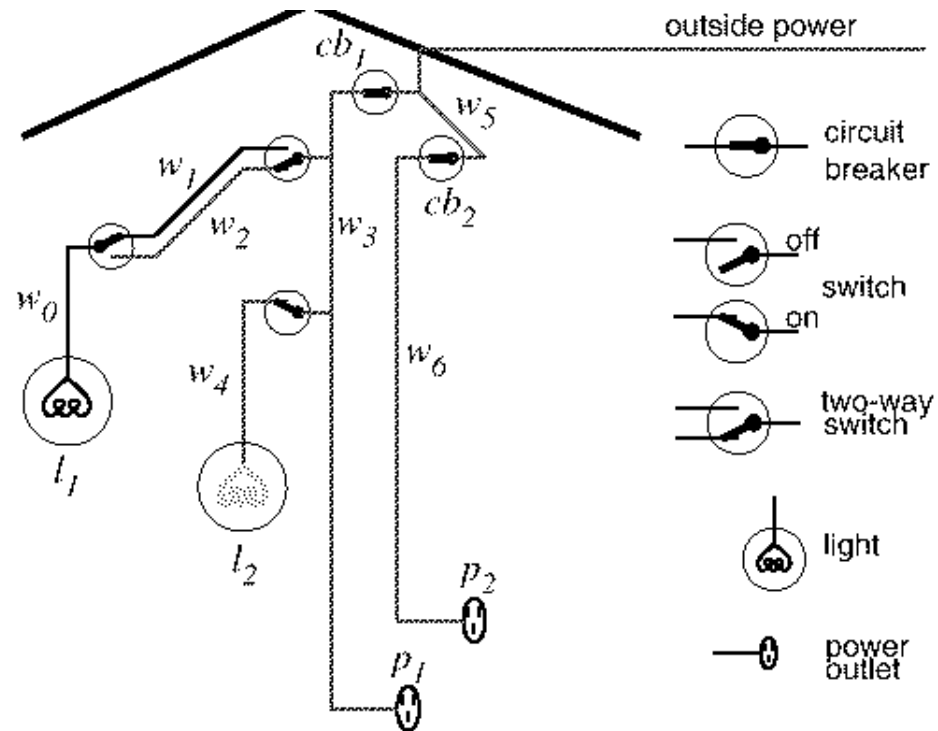
- Entradas del robot repartidor:
  - *Conocimiento previo*: sus capacidades, objetos existentes, plano de la oficina
  - *Experiencias anteriores*: qué acciones son útiles y cuándo, efectos de sus acciones sobre su posición y su entorno
  - *Objetivos*: qué repartir y cuándo
  - *Valores*: para ordenar objetivos
  - *Observaciones*: sobre su entorno

# El robot repartidor

- Tareas del robot repartidor:
  - Determinar la posición del despacho de una persona, del café, ...
  - Buscar un camino entre dos posiciones
  - Planificar cómo realizar varias tareas
  - Conjeturar la posición de una persona
  - Tomar decisiones con incertidumbre
  - Aprender de la experiencia
  - Percibir el mundo: saber dónde está, evitar obstáculos, ...

# El S.E. electricista

- El sistema eléctrico (Poole-98 p. 16)



## El S.E. electricista

- Entradas del S.E. electricista:
  - *Conocimiento previo*: funcionamiento de las luces y las conexiones, síntomas de fallos, información aportada por los tests, efectos de las reparaciones
  - *Experiencias anteriores*: datos de casos anteriores
  - *Objetivo*: arreglar el sistema
  - *Valores*: para decidir entre reparar o sustituir componentes
  - *Observaciones*: síntomas del sistema

## El S.E. electricista

- Tareas del S.E. electricista:
  - Determinar los efectos de los fallos y las intervenciones
  - Buscar en el espacio de posibles fallos
  - Explicar su razonamiento al usuario
  - Derivar posibles causas de fallos
  - Planificar realización de pruebas o tratamientos
  - Conjeturar problemas usando conocimiento por defecto
  - Razonar con incertidumbre y conocimiento incompleto
  - Aprender cómo asociar síntomas con fallos, efectos de los tratamientos, confianza de los tests

# El infobot

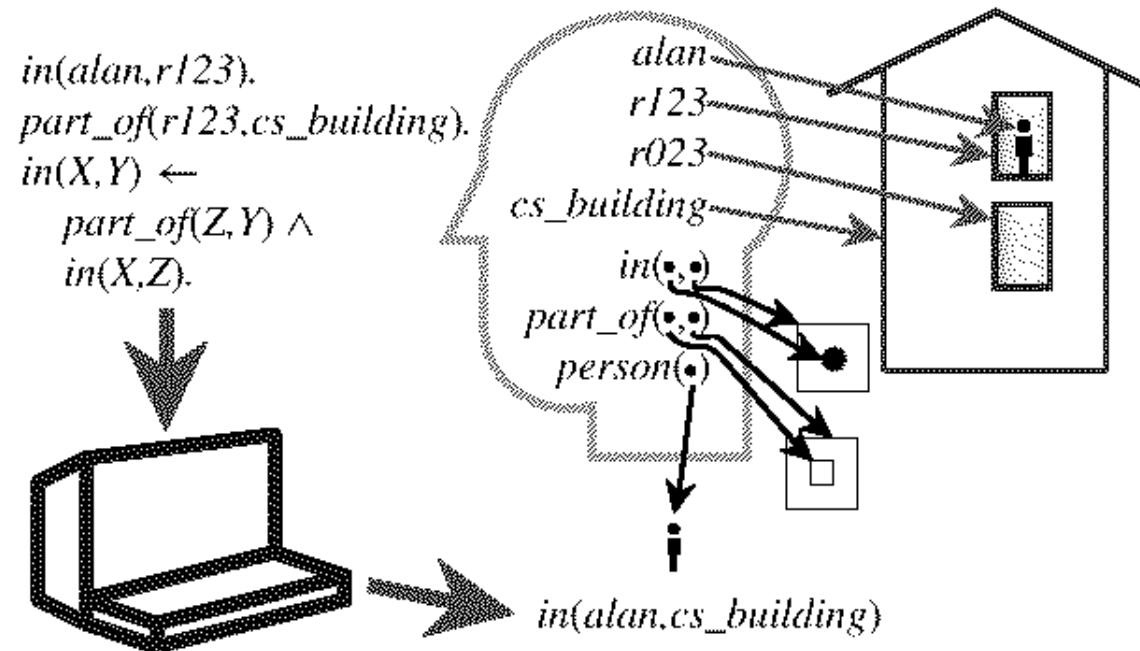
- **Interacción del infobot con su entorno informático:**
  - Recibe preguntas en un lenguaje de alto nivel
  - Busca la información relevante
  - Presenta la información de manera legible
- **Entradas del infobot:**
  - *Conocimiento previo:* significado de las palabras, tipos de fuentes de información, cómo acceder a la información
  - *Experiencias anteriores:* dónde puede obtenerse información, la velocidad relativa de los servidores, preferencias del usuario
  - *Objetivo:* la información buscada
  - *Valores:* para decidir entre el volumen y la calidad de la información
  - *Observaciones:* qué información hay en el sitio actual, qué enlaces hay

# El infobot

- Tareas del infobot:
  - Derivar información que sólo está implícita en una base de conocimientos
  - Interactuar en lenguaje natural
  - Buscar entre bases de conocimientos la información
  - Representar el conocimiento eficientemente
  - Explicar el razonamiento justificativo de las respuestas
  - Tomar decisiones con conocimiento incompleto o contradictorio
  - Razonar por defecto sobre dónde encontrar información
  - Decidir entre calidad de la información y coste
  - Aprender preferencias del usuario y fuentes de información

## Uso de un SRR

- Papel de la semántica en SRR (Poole-98 p. 26)



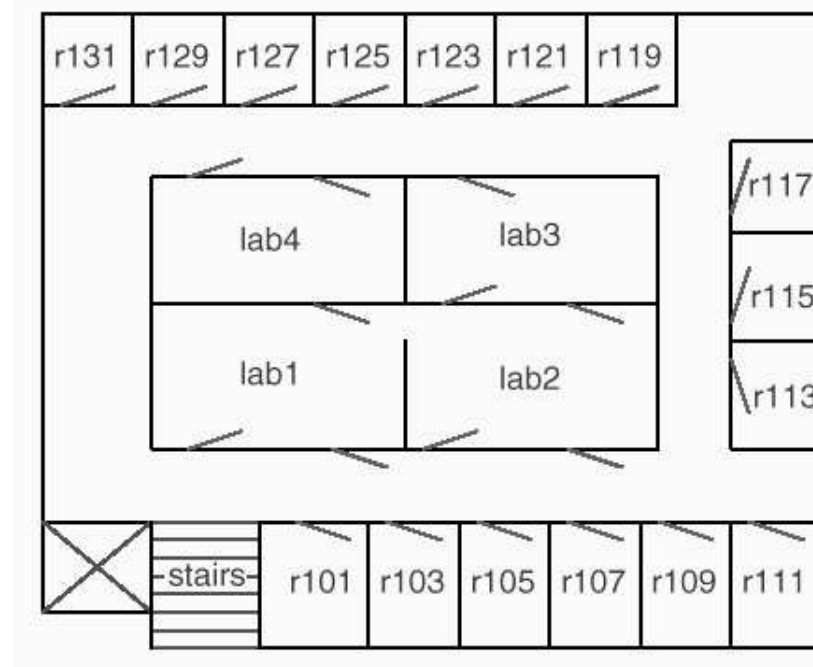


## Uso de un SRR

- Elegir el dominio de la tarea o el mundo que se desea representar (*interpretación deseada*)
- Asociar una constante del lenguaje a cada individuo del mundo que se representa
- Asociar un símbolo de predicado del lenguaje a cada relación que se desee representar
- Decirle al SRR las cláusulas que son verdaderas en la interpretación deseada (*axiomatización del dominio*)
- Preguntar al SRR e interpretar las respuestas

# Dominio del robot repartidor

- El mundo del robot repartidor (Poole-98 p. 14)



# Dominio del robot repartidor

- **Constantes:** h101, h103, h105, h107, h109, h111, h127, h129, h131
- **Relación:** `vecina_izquierda(H1,H2)` que es verdad si la habitación H1 es la vecina izquierda de la habitación H2

- **Base de conocimiento**

```
vecina_izquierda(h101,h103).  
vecina_izquierda(h103,h105).  
vecina_izquierda(h105,h107).  
vecina_izquierda(h107,h109).  
vecina_izquierda(h109,h111).  
vecina_izquierda(h131,h129).  
vecina_izquierda(h129,h127).  
vecina_izquierda(h127,h125).
```

# Dominio del robot repartidor

- Consultas

```
?- vecina_izquierda(h105,h107). => Yes
?- vecina_izquierda(h106,h107). => No
?- vecina_izquierda(X,h107).    => X = h105
?- vecina_izquierda(h105,X).    => X = h107
?- vecina_izquierda(h106,X).    => No
```

- Relación definida `vecina_derecha(H1,H2)` que es verdad si la habitación H2 es la vecina derecha de la habitación H1

- Definición

```
vecina_derecha(H1,H2) :- vecina_izquierda(H2,H1).
```

- Consulta

```
?- vecina_derecha(h105,X). => X = h103
```

# Dominio del robot repartidor

- Relación definida `vecina(H1,H2)` que es verdad si la habitación H1 es vecina de la habitación H2

- Definición

```
vecina(H1,H2) :- vecina_derecha(H1,H2).  
vecina(H1,H2) :- vecina_izquierda(H1,H2).
```

- Consulta

```
?- vecina(X,h105). => X = h107 ; X = h103
```

- Relación definida `dos_a_la_derecha(H1,H2)` que es verdad si la habitación H1 está dos a la derecha de la habitación H2

- Definición

```
dos_a_la_derecha(H1,H2) :- vecina_derecha(H1,H), vecina_derecha(H,H2).
```

- Consulta

```
?- dos_a_la_derecha(X,h105). => X = h109
```

# Dominio del robot repartidor

- Relación definida  $a\_la\_izquierda(H1,H2)$  que es verdad si la habitación H1 está a la izquierda de la habitación H2

- Definición recursiva

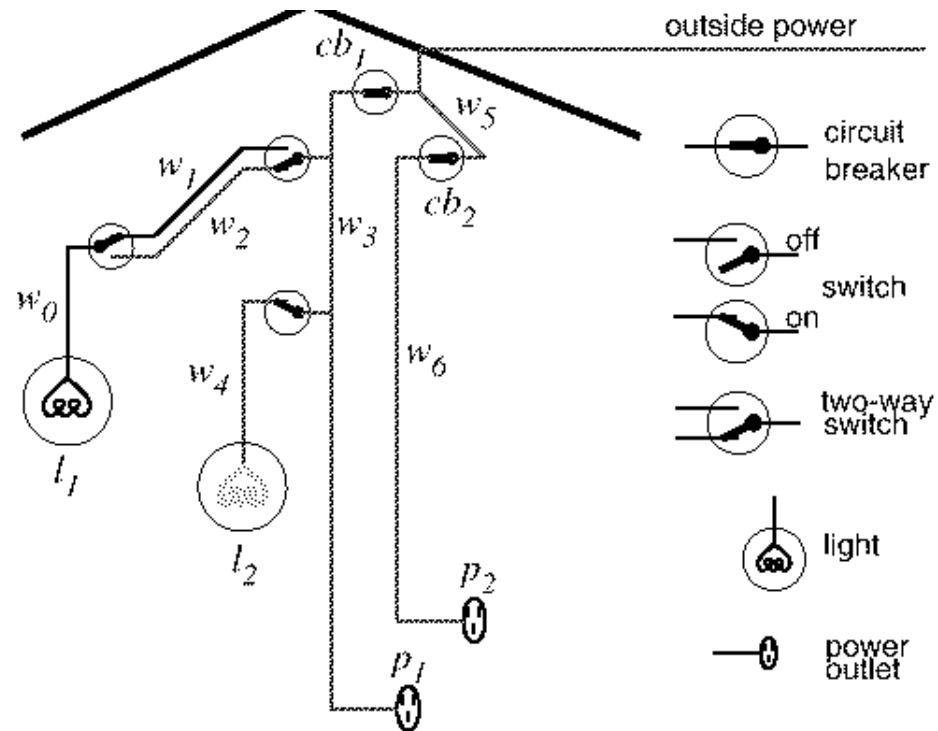
```
a_la_izquierda(H1,H2) :-  
    vecina_izquierda(H1,H2).  
a_la_izquierda(H1,H2) :-  
    vecina_izquierda(H1,H),  
    a_la_izquierda(H,H2).
```

- Consulta

```
?- a_la_izquierda(X,h105).  
X = h103 ;  
X = h101 ;  
No
```

# Dominio del sistema eléctrico

- El sistema eléctrico (Poole-98 p. 16)



# Dominio del sistema eléctrico

- Constantes:
  - Luces: l1, l2
  - Interruptores: i1, i2, i3
  - Cortacircuitos: cc1, cc2
  - Cables: c1, c2, c3, c4, c5, c6
  - Enchufes: e1, e2
  - Toma de corriente: entrada



# Dominio del sistema eléctrico

- Predicados:

- luz(L) es verdad si L es una luz
- abajo(I) es verdad si el interruptor I está hacia abajo
- arriba(I) es verdad si el interruptor I está hacia arriba
- esta\_bien(X) es verdad si la luz o el cortocircuito X está bien
- conectado(D1,D2) es verdad si los dispositivos D1 y D2 está conectados (de forma que puede fluir la corriente eléctrica de D2 a D1)
- tiene\_corriente(D) es verdad si el dispositivo D tiene corriente
- esta\_encendida(L) es verdad si la luz L está encendida

# Dominio del sistema eléctrico

- Base de conocimiento del sistema eléctrico

```
luz(l1). luz(l2).
```

```
abajo(i1).
```

```
arriba(i2). arriba(i3).
```

```
esta_bien(l1). esta_bien(l2). esta_bien(cc1). esta_bien(cc2).
```

```
conectado(l1,c0).
```

```
conectado(c0,c1) :- arriba(i2).
```

```
conectado(c0,c2) :- abajo(i2).
```

```
conectado(c1,c3) :- arriba(i1).
```

```
conectado(c2,c3) :- abajo(i1).
```

```
conectado(l2,c4).
```

```
conectado(c4,c3) :- arriba(i3).
```

```
conectado(e1,c3).
```

```
conectado(c3,c5) :- esta_bien(cc1).
```

```
conectado(e2,c6).
```

```
conectado(c6,c5) :- esta_bien(cc2).
```

```
conectado(c5,entrada).
```

# Dominio del sistema eléctrico

```
tiene_corriente(D) :-  
    conectado(D,D1),  
    tiene_corriente(D1).  
tiene_corriente(entrada).
```

```
esta_encendida(L) :-  
    luz(L),  
    esta_bien(L),  
    tiene_corriente(L).
```

## ● Consultas

```
?- tiene_corriente(D). => D = c2 ; D = l2 ; D = c4 ; D = e1 ; D = c3 ;  
                        D = e2 ; D = c6 ; D = c5 ; D = entrada  
?- esta_encendida(X). => X = l2
```

## Bibliografía

- Poole, D.; Mackworth, A. y Goebel, R. *Computational Intelligence (A Logical Approach)* (Oxford University Press, 1998)
  - Cap. 1: “Computational intelligence and knowledge”
  - Cap. 2: “A representation and reasoning system”
  - Cap. 3: “Using definite knowledge”
- Russell, S. y Norvig, P. *Inteligencia artificial (Un enfoque moderno)* (Prentice–Hall Hispanoamericana, 1996)
  - Cap. 1: “Introducción”
  - Cap. 2: “Agentes inteligentes”
  - Cap. 6: “Agentes que razonan de manera lógica”
  - Cap. 10: “Sistemas de razonamiento lógico”