



# Desarrollo Basado en Agentes

## Tema 2. Autonomía (II)

*Luis Castillo Vidal*

*Dpto Ciencias de la Computación e I.A.  
2016-2017*



# Esquema del tema

- Motivación
- Modelo básico de agentes que actúan
- Agentes deductivos
- Agentes de razonamiento práctico
- Agentes reactivos e híbridos



# Agentes deductivos

- Agentes de razonamiento simbólico
  - a. El agente contiene un **modelo simbólico**
    - i. De sí mismo (capacidades)
    - ii. Del entorno → ¿de otros agentes?
  - b. Las decisiones están dirigidas por procesos de **razonamiento simbólico**
  - c. Comunmente son modelos y razonamiento basados en lógica (predicados, proposicional)
- Independientes del lenguaje de programación

# Agentes deductivos

- “Teletransportables”: si un agente es su KB, entonces ésta se puede transmitir (total o parcialmente) de un agente a otro
  - a. Agentes persistentes
  - b. Clonar agentes
  - c. Inyección de conocimiento

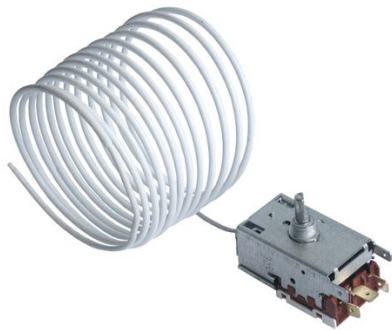


# Agentes deductivos

- Idea general
  - a. Sea  $KB = H \cup R$ 
    - i.  $H$  (hechos) =  $\{h1, h2, \dots, hn\}$ 
      - 1.  $hi = literal(arg1, arg2, \dots, argn) \rightarrow padre(X, Y)$
      - 2. La ontología
        - a. Las clases de objetos, sus propiedades y sus relaciones
      - 3. El estado de las cosas (Instancias)  $\rightarrow padre(Luis, Ana)$
    - ii.  $R$  (reglas) =  $\{r1, r2, \dots, rn\}$ 
      - 1. Operadores lógicos  $\wedge \vee \neg \rightarrow$
      - 2. Las acciones y capacidades del agente
  - b. El problema de actuar para conseguir unos objetivos se reduce a la demostración de teoremas
    - i.  $KB \vdash \text{objetivos}$
    - ii.  $\vdash$  mecanismos lógicos de inferencia
    - iii. La demostración  $\rightarrow$  acción/plan-de-acción

# Agentes deductivos

- Idea general



## KB

$H = \{T(17, s_0), C(15, s_0)\}$  // Temperatura y consigna

$R = \{\text{ON}(S, S') :: T(X, S) \wedge C(Y, S) \wedge \text{Mayor}(X, Y) \rightarrow T(X-1, S') \text{ // Entorno discreto}$

$\text{OFF}(S, S') :: T(X, S) \wedge C(Y, S) \wedge \text{MenorOlg}(X, Y) \rightarrow T(X, S')\}$

## Objetivo

$O = T(15, S'')$

## Demostración $KB \vdash O$

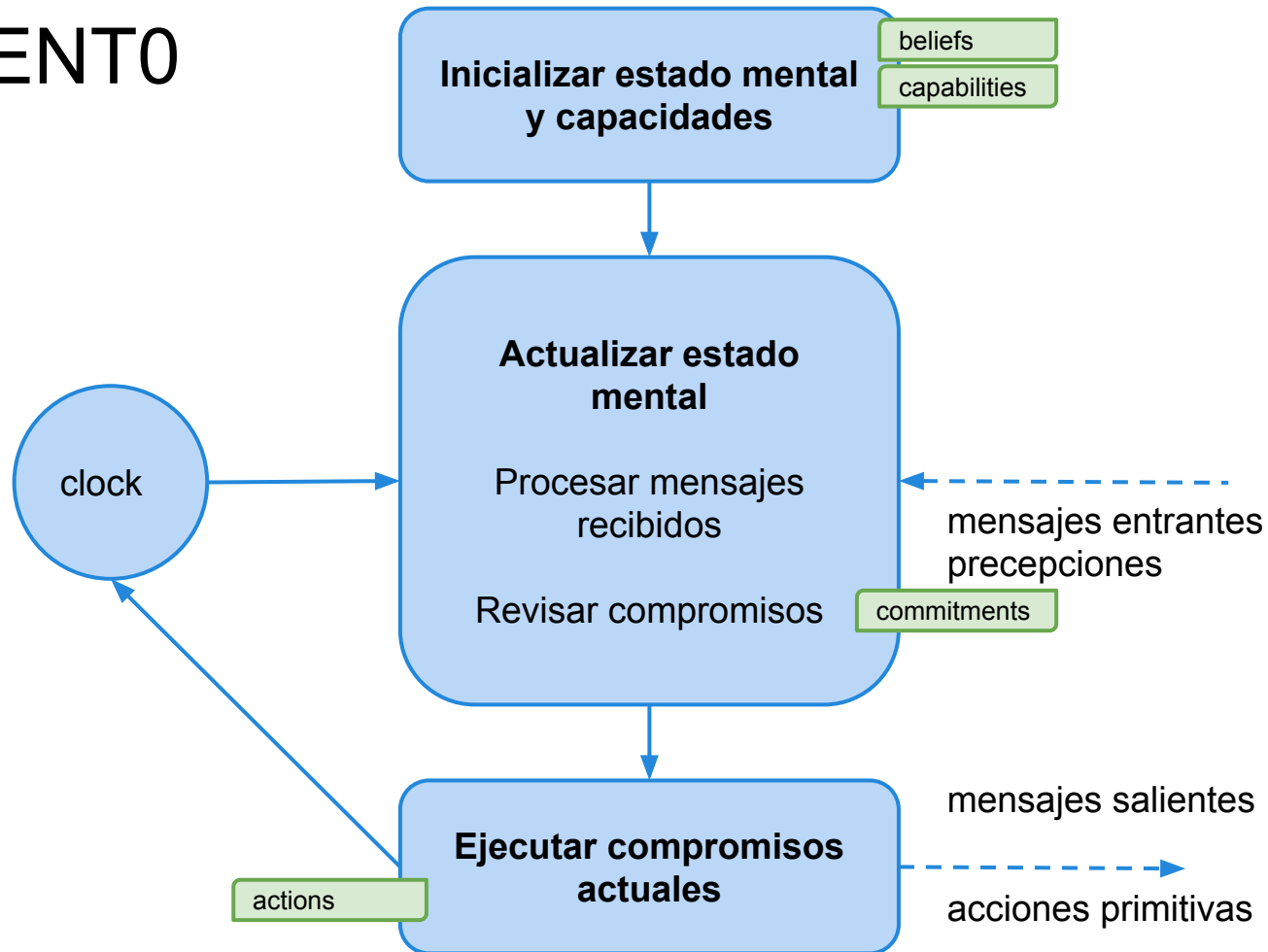
$\{T(17, s_0), C(15, s_0)\} \xrightarrow{\text{ON}} \{T(16, s_1), C(15, s_1)\} \xrightarrow{\text{ON}} \{T(15, s_2), C(15, s_2)\} \xrightarrow{\text{OFF}}$

## Actuación

$\{\text{On}(s_0, s_1), \text{On}(s_1, S''=s_2)\}$

# Agentes deductivos

- AGENT0



beliefs	commitments
capabilities	actions

# Agentes deductivos

- AGENT0 (+ info [Shoham AOP](#))
  - a. Estados mentales del agente
    - i. **Beliefs:** estado del mundo, *mental conditions* (MTC)
      1. (AND (OR (NOT ... (*tiempo* (*predicado argumentos*)))
      2. (and (0 (abierta puerta1))(1 (posicion robot2 puerta1)))

Notación prefija LISP

3+4  $\equiv$  (+ 3 4)    abierta(puerta1)  $\equiv$  (abierta puerta1)

- ii. **Capabilities:** Las acciones que puede realizar el agente y sus precondiciones

(CAN (DO *tiempo accionprimitiva*) *mtc-precondition*)

(CAN (DO 10:00 abrirpuerta) (and (10:00 (posicion myself ?p))  
(10:00 (posicion puerta ?p))))

(CAN (DO ?anytime shutdown) true)

(CAN (DO ?t moveSE) (and (?t (bateria-nivel ?n))  
(> ?n 1)))

?p  $\rightarrow$  variable lógica libre para ser instanciada



# Agentes deductivos

## ● AGENT0

### a. Estados mentales del agente

- i. **Commitments:** Reglas que expresan cuándo el agente se compromete a algo con otro (*agente2*), de motu propio o porque se lo pide otro agente (*msg*)

1. (COMMIT *msg mtc* *agente2* (DO *tiempo accionprimitiva*))

(COMMIT true (and (?t (bateria ?n)) (> ?n 10)) controler (DO ?t2 abrirpuerta))

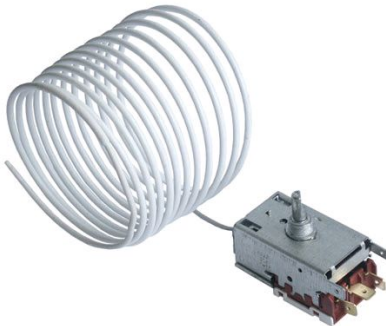
2. Cada compromiso que se dispare modifica las creencias (*tiempo<sup>c</sup>* (CMT *agente2* (DO *tiempo<sup>a</sup>* *aprim*))) → Beliefs

- ii. **Actions:** lo que va a hacer el agente

1. Acciones primitivas (DO *tiempo accionprimitiva*)
  - a. (DO 10:00 abrirpuerta)
2. Acciones comunicativas
3. Acciones condicionadas

# Agentes deductivos

- AGENT0



Agent Termostato

(TIMEGRAIN m)

(CAPABILITIES

(CAN (DO ?t ON) true)

(CAN (DO ?t OFF) true))

(BELIEFS

(0 (T 17)) (0 (C 15)) (0 (E OFF))

(0 (CAN (DO ?t ON) true)

(0 (CAN (DO ?t OFF) true)

(0 (CMT Timer) (DO 22:00 OFF))

)

(COMMITMENTS

...

)

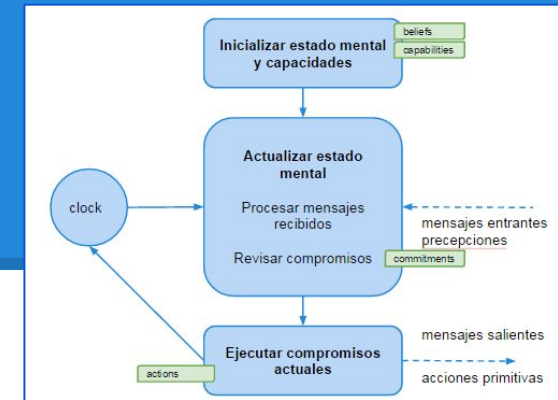
Resolución temporal  
s | m | h | d | y

Variables lógicas ?t

**Introspección** de las  
capabilities y commitments.  
Solo se muestran los  
hechos y los commitments  
por simplicidad

# Agentes deductivos

## ● AGENT0



Agent Termostato v0

```
(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (0 (T 17)) (0 (C 15))(0 (E OFF))
)
(COMMITMENTS )
```

**Revisar Compromisos en CLOCK = 0**

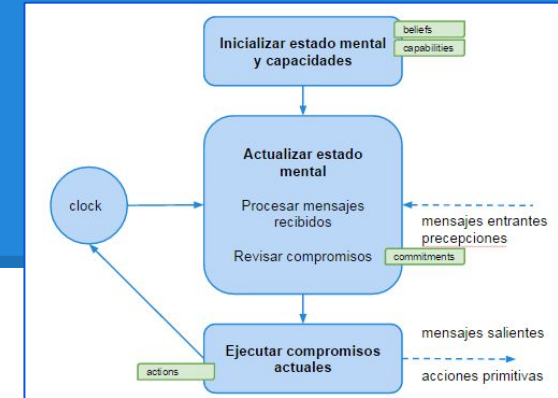
null

**Ejecutar Compromisos para CLOCK = 0**

null

# Agentes deductivos

## ● AGENT0



Agent Termostato v0

```
(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (1 (T 18)) (1 (C 15))(1 (E OFF))
)
(COMMITMENTS )
```

**Revisar Compromisos en CLOCK = 1**

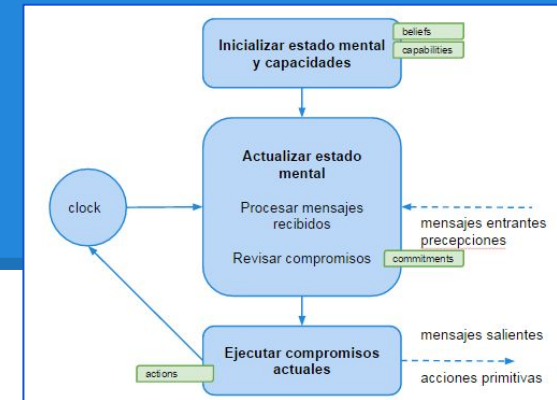
null

**Ejecutar Compromisos para CLOCK = 1**

null

# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (0 (T 17)) (0 (C 15))(0 (E OFF))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO ?t ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO ?t OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

### Consulta estado cognitivo

(tiempo (pred arg))

(tiempo (CMT agente accion))

(tiempo (CAN accion mtc))

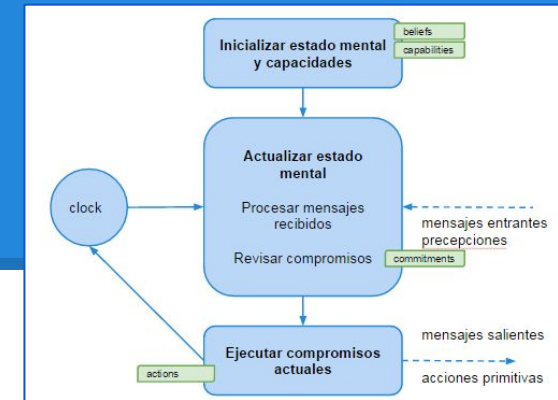
→ Estado

→ Commitment

→ Capability

# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (0 (T 17)) (0 (C 15))(0 (E OFF))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO ?t ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO ?t OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Revisar Compromisos en CLOCK = 0**

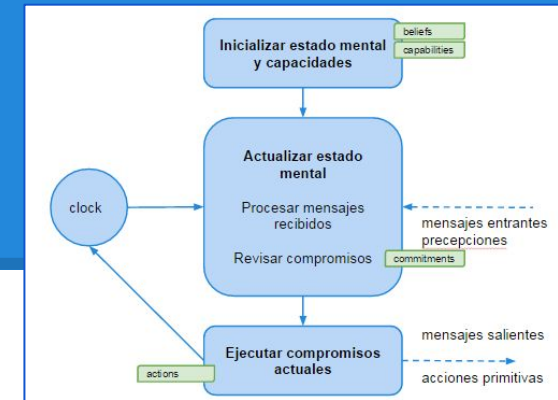
```

(0 (CMT Termostato (DO 0 ON)))
(0 (CMT Timer (DO 23:59 OFF)))
  
```

→ BELIEFS

# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (0 (T 17)) (0 (C 15))(0 (E OFF)) (0 (CMT Termostato (DO 0 ON)))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO ?t ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO ?t OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Revisar Compromisos en CLOCK = 0**

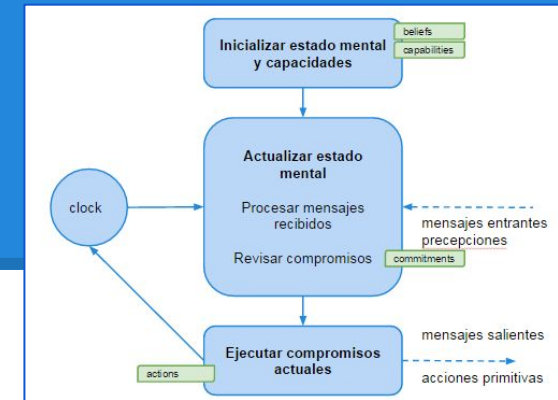
```

(0 (CMT Termostato (DO 0 ON)))
(0 (CMT Timer (DO 23:59 OFF)))
  
```

→ BELIEFS

# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

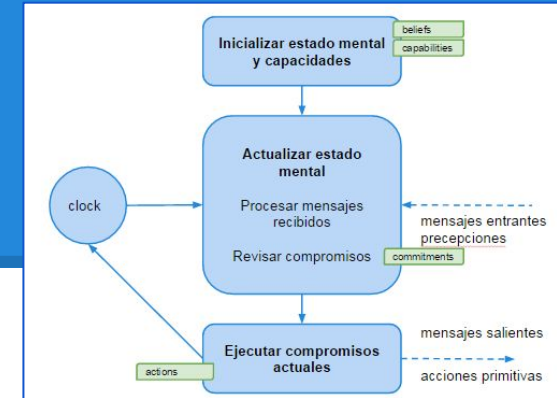
(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (0 (T 17)) (0 (C 15))(0 (E OFF)) (0 (CMT Termostato (DO 0 ON)))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO 0 ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO 23:59 OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Ejecutar Compromisos** para CLOCK = 0  
(0 (CMT termostato (DO 0 ON)))



# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (0 (T 17)) (0 (C 15))(0 (E OFF))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO 0 ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO 0 OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Ejecutar Compromisos** para CLOCK = 0

(0 (CMT termostato (DO 0 ON)))

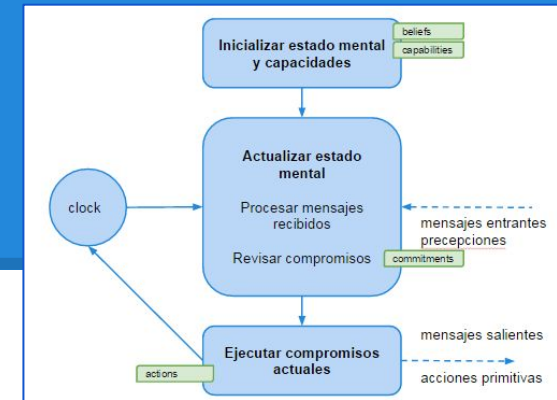
El termostato se activa  
Aún no hemos actualizado las percepciones

- AGENT0



# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

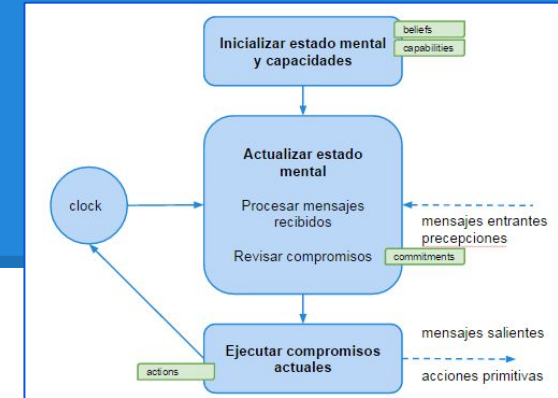
(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (1 (T 16)) (1 (C 15))(1 (E ON))(1 (CMT Termostato (DO 1 ON)))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO ?t ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO ?t OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Revisar Compromisos en CLOCK = 1**

(1 (CMT termostato (DO 1 ON)))  
→ BELIEFS

# Agentes deductivos

## ● AGENT0



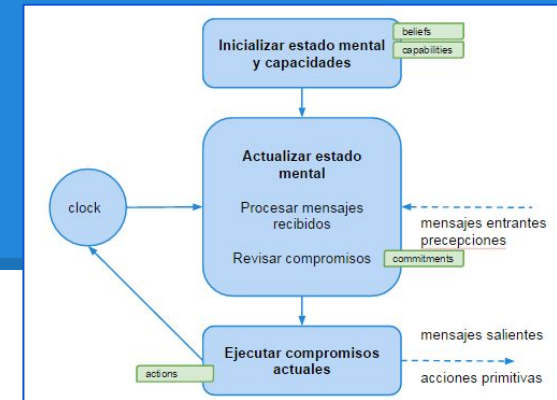
Agent Termostato v1

```

(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (1 (T 16)) (1 (C 15))(1 (E ON))(1 (CMT Termostato (DO 1 ON)))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO 1 ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO 1 ON))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Ejecutar Compromisos** para CLOCK = 1  
(1 (CMT termostato (DO 1 ON)))

- AGENT0



# Agent Termostato v1

```
(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (1 (T 16)) (1 (C 15))(1 (E ON))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (
  (COMMIT true true Timer (DO 23:59 OFF))
)
)
```

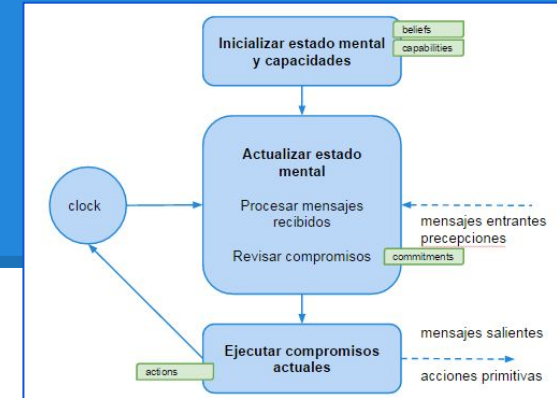
### Ejecutar Compromisos para CLOCK = 1

(1 (CMT termostato (DO 1 ON)))

El termostato se activa  
Aún no hemos actualizado las percepciones

# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

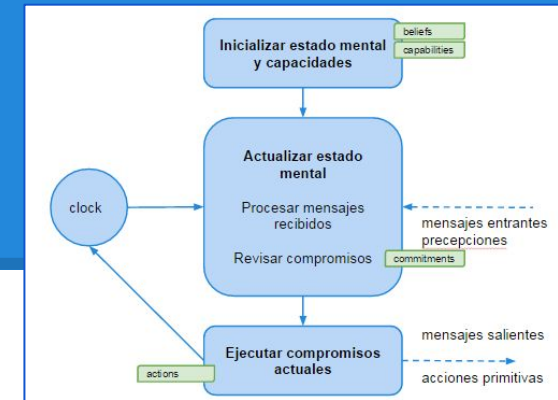
(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (2 (T 15)) (2 (C 15))(2 (E ON))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO ?t ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO ?t OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Revisar Compromisos en CLOCK = 2**

(2 (CMT termostato (DO 2 OFF)))  
→ BELIEFS

# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (2 (T 15)) (2 (C 15))(2 (E ON))(2 (CMT Termostato (DO 2 OFF)))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO ?t ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO ?t OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

**Revisar Compromisos en CLOCK = 2**

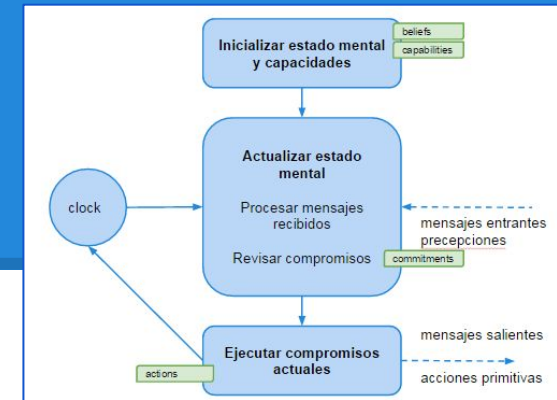
(2 (CMT termostato (DO 2 OFF)))  
→ BELIEFS





# Agentes deductivos

## ● AGENT0



Agent Termostato v1

```

(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (2 (T 15)) (2 (C 15))(2 (E ON))
  (0 (CMT Timer (DO 23:59 OFF)))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO 2 ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO 2 OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
  
```

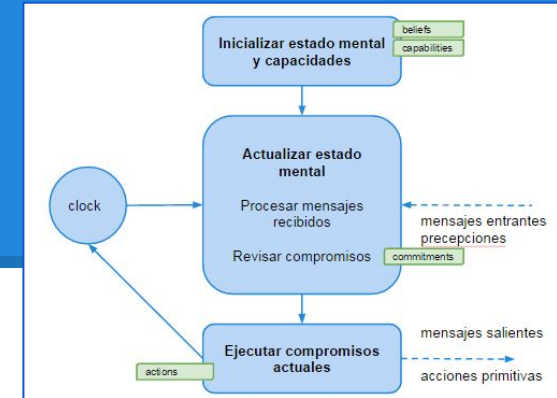
**Ejecutar Compromisos** para CLOCK = 2

(2 (CMT termostato (DO 2 OFF)))

El termostato se desactiva  
Aún no hemos actualizado las percepciones

# Agentes deductivos

- AGENT0



Agent Termostato v1

```
(TIMEGRAIN m)
(CAPABILITIES
  (CAN (DO ?t ON) true)
  (CAN (DO ?t OFF) true))
(BELIEFS
  (3 (T 15)) (3 (C 15))(3 (E OFF))
  (0 (CMT Timer (DO 23:59 OFF))))
)
(COMMITMENTS
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (> ?TE ?CO)) Termostato (DO ?t ON))
  (COMMIT true (and (?t (T ?TE)) (?t (C ?CO)) (<= ?TE ?CO)) Termostato (DO ?t OFF))
  (COMMIT true true Timer (DO 23:59 OFF))
)
)
```

# Agentes deductivos

- AGENT0

- a. Acciones primitivas

- (DO *tiempo accionprimitiva*)

- b. Acciones comunicativas

- i. (REQUEST *tiempo agente2 accion*)

FIPA ACL

- ii. (INFORM *tiempo agente2 mtc*)

- c. Acciones condicionadas

- i. (IF *mtc accion*)

- d. Condiciones sobre los mensajes recibidos

- (COMMIT *msg mtc agente2 (DO tiempo accionprimitiva)*)

- i. (*agentesender REQUEST accion*)

- ii. (*agentesender INFORM mtc*)

# Agentes deductivos

- AGENT0

- a. Agentes “Termostato”, “Timer”, “User”

Agent Termostato v3

(TIMEGRAIN m)

(CAPABILITIES

(CAN (DO ?t ON) true)

(CAN (DO ?t OFF) true))

(CAN (REQUEST ?t ?X ?A) true)

(CAN (INFORM ?t ?X ?mtc) true)

(BELIEFS

(3 (T 15)) (3 (C 15))(3 (E OFF))

)

(COMMITMENTS

(COMMIT true (and (B (and (?t (T ?TE))) (?t (C ?CO)))) (> ?TE ?CO)) Termostato (DO ?t ON))

(COMMIT true (and (B (and (?t (T ?TE))) (?t (C ?CO)))) (<= ?TE ?CO)) Termostato (DO ?t OFF))

(COMMIT (Timer REQUEST (DO ?t OFF)) true Timer (DO ?t OFF))

(COMMIT (?anyagent REQUEST (INFORM ?t ?anyagent ?mtc)) true ?anyagent (INFORM ?t ?anyagent ?mtc))

(COMMIT true (and (?t (E OFF)) (?t (T ?TE))) (?t (C ?CO)) (> ?TE ?CO)) User (INFORM ?t User (?t (E BAD)))

)

# Agentes deductivos

- AGENT0: Buenas propiedades
  - Consistencia de las creencias (beliefs)
    - No debe haber hechos inconsistentes
      - (BELIEFS (0 (not (abierta puerta))) (0 (abierta puerta)))
    - Y si las hay ¿cuál eliminar?
  - Actuación de buena fé
    - El agente no puede comprometerse a hacer algo que no está entre sus capacidades
    - El agente no puede informar sobre la veracidad de algo que desconoce

# Agentes deductivos

- El agente no puede comprometerse a hacer algo que no está entre sus capacidades

(COMMIT (?Ag REQUEST (DO ?t ?action)) (and [0 (CAN (DO ?t ?action) ?mtc] ?mtc) ?Ag (DO ?t ?action))

# Agentes deductivos

- El agente no puede informar sobre la veracidad de algo que desconoce

(COMMIT (?A REQUEST (INFORM ?t ?A2 ?mtc)) (and ?mtc (0 (CAN (INFORM ?AT ?AA ?AC)))) ?A (INFORM ?t ?A2 ?mtc))

# Agentes deductivos

- AGENT0: Buenas propiedades
  - Persistencia del estado mental
    - **Beliefs**: actualizar y eliminar aquellos hechos que dejan de ser ciertos
    - **Capabilities**: inicialmente son estáticas, pero en agentes avanzados podrían variar (aprender nuevas capacidades o eliminar aquellas obsoletas).
      - “Ya sé Kung-Fu” → ¡no tan futurista!
    - **Commitments**: persisten hasta que son completadas.
      - En algunos modelos de agentes, el agente que hace un REQUEST puede hacer un UNREQUEST cuando ya lo no necesita, liberando así al agente que se comprometió



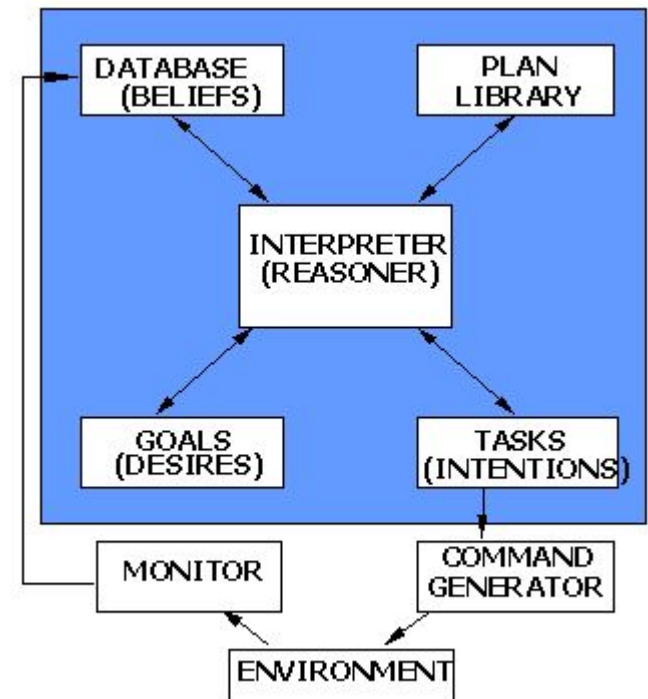


# Agentes de razonamiento práctico

- Agentes deliberativos: Razonan sobre **secuencias de acciones** y su efecto en el entorno
- Asociada a agentes (intención)
- SIEMPRE sobre un modelo deductivo
- Entradas
  - i. Objetivos
  - ii. Estado actual → (i+ii) Problema
  - iii. Acciones → (iii) Dominio
- Salidas
  - i. Plan

# Agentes de razonamiento práctico

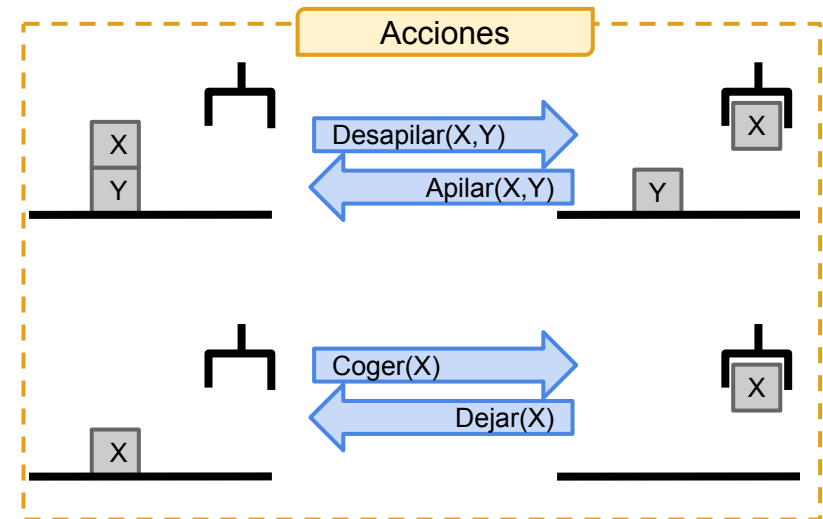
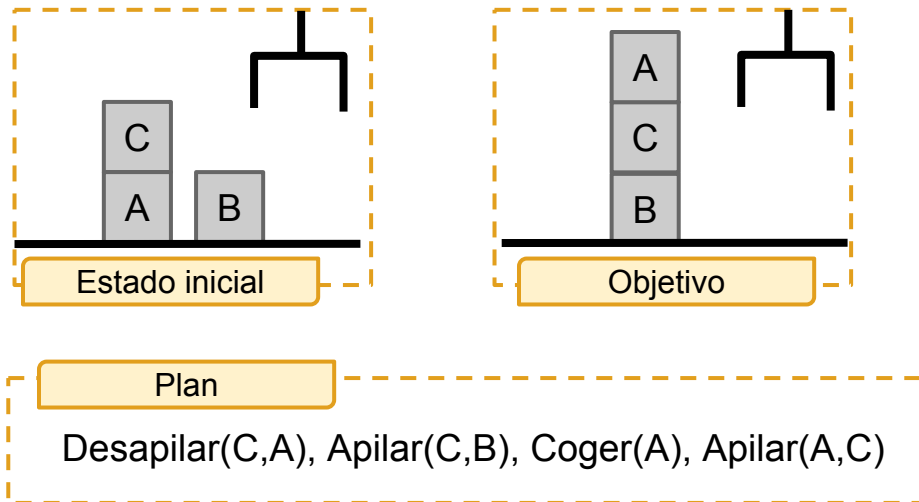
- Agentes BDI (+info [Wikipedia](#))
  - Beliefs
  - Desires
  - Intentions



# Agentes de razonamiento práctico

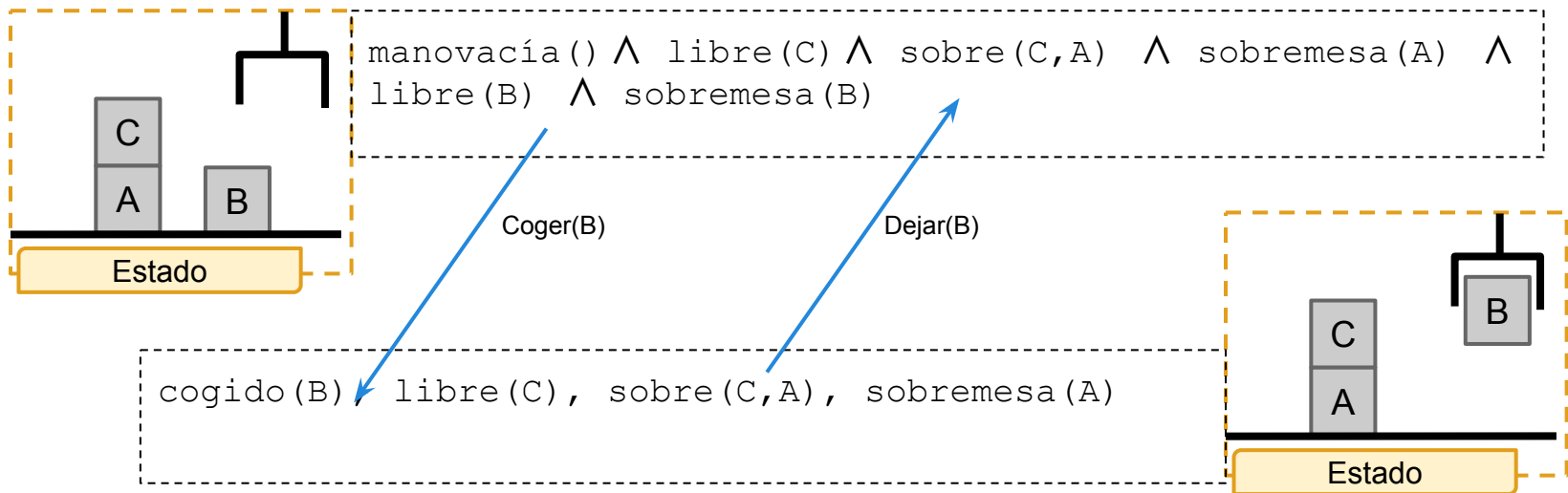
- Agentes que planifican  $\subset$  BDI

- a. El mundo de bloques
  - i. Uno (o más robots)
  - ii. Una serie de bloques
  - iii. Una superficie
- b. Un problema de ejemplo



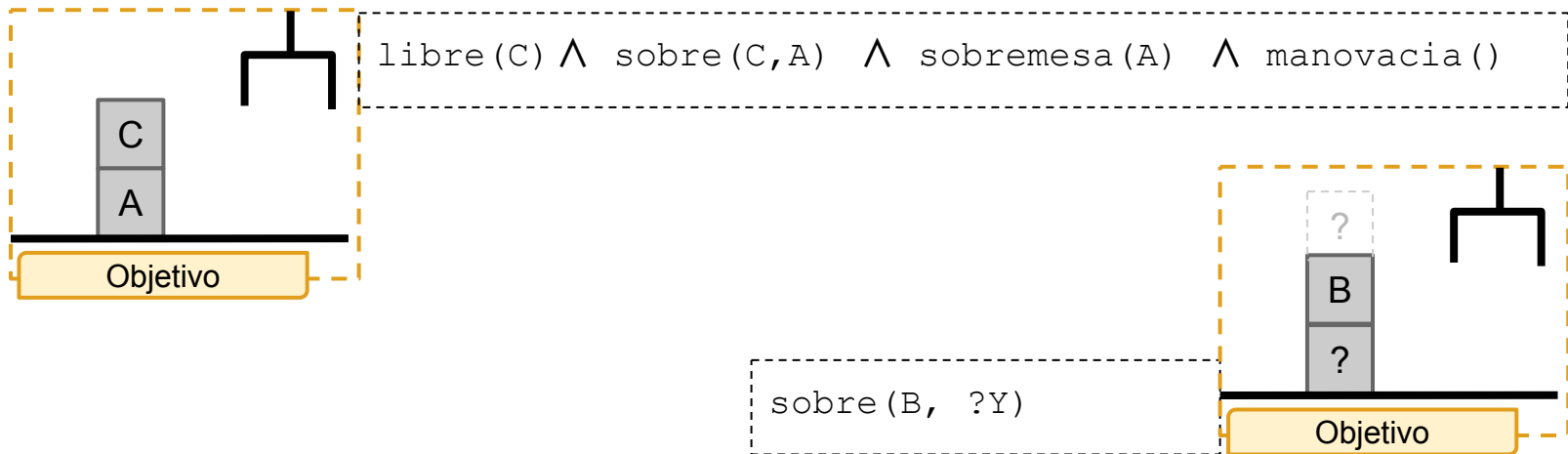
# Agentes de razonamiento práctico

- Agentes que planifican
  - a. Lenguaje de representación
    - i. **De estados.** Suele estar basado en la lógica de predicados.
      1. Completitud → **Hipótesis de mundo cerrado (CWA)**
      2. En problemas **con percepción real CWA ¡no existe!**



# Agentes de razonamiento práctico

- Agentes que planifican
  - a. Lenguaje de representación
    - i. **De objetivos.** Suele estar basado en la lógica de predicados. Puede contener variables o estar parcialmente definido (conjunto de estados)



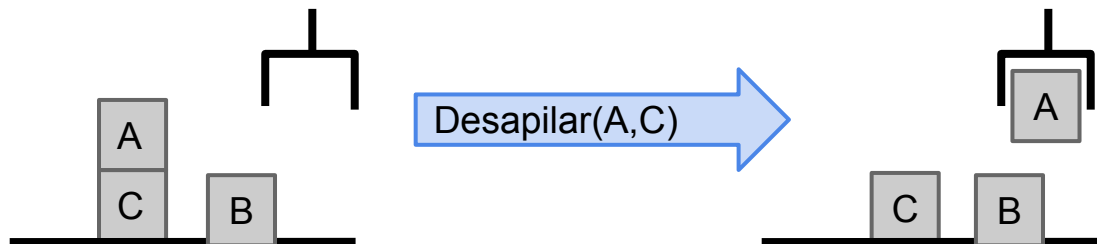
# Agentes de razonamiento práctico

- Agentes que planifican
  - a. Lenguaje de representación
    - i. **De acciones.** Suele estar basado en la lógica de predicados.
      - Calculo situacional (lógica predicados)

Axioma Desapilar ?X de ?Y, cambia del estado S al S'

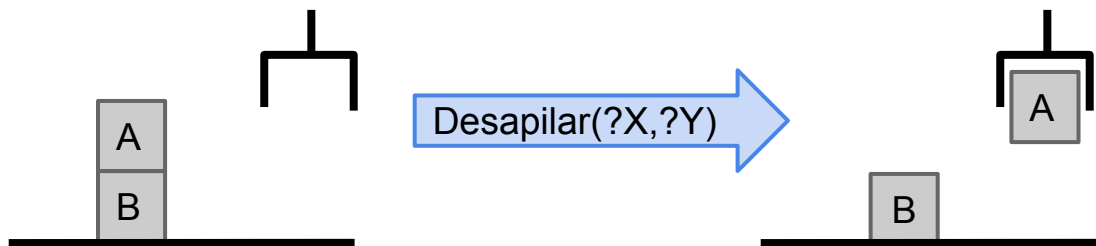
$$\text{manovacia}(S), \text{libre}(?X, S), \text{sobre}(?X, ?Y, S) \rightarrow \neg \text{manovacia}(S') \wedge \neg \text{libre}(?X, S') \wedge \neg \text{sobre}(?X, ?Y, S') \wedge \text{cogido}(?X, S') \wedge \text{libre}(?Y, S')$$

- Problema del marco  $\rightarrow$  Intratabilidad



# Agentes de razonamiento práctico

- Agentes que planifican
  - a. Lenguaje de representación
    - i. **De acciones.** Suele estar basado en la lógica de predicados.
      - Planificador STRIPS (basado en la lógica)
      - Listas de precondition, supresión y adición → Probl. del marco
        - Con variables (lógica de predicados)
        - Sin variables (lógica proposicional)



## Acción Desapilar(?X,?Y)

P: manovacia(), libre(?X), sobre(?X,?Y)  
S: manovacia(), libre(?X), sobre(?X, ?Y)  
A: cogido(?X), libre(?Y)

# Agentes de razonamiento práctico

- PDDL (Planning Domain Description Language)

## domain.pddl

Acciones posibles

```
;; Ontología
(:types Block - Object)
(:predicates (libre ?X - Block) ... )
(:functions (peso ?X - Block) ... )

;; Capacidades
(:action Desapilar
:parameters (?X ?Y - Block)
:condition (and (manovacia) (libre ?X)
               (sobre ?X ?Y))
:duration (= ?duration 25)
:effects (and (not (manovacia))
              (not (libre ?X))
              (not (sobre ?X ?Y))
              (cogido ?X) (libre ?Y)))
```

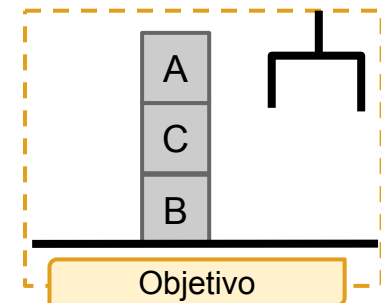
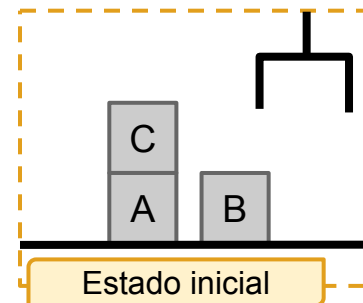
## problem.pddl

Percepción y  
objetivo

```
;; Instancias de la ontología
(:objects A, B, C - Block)

;; Estado inicial (Creencias)
(:init (and (libre C) (sobre C A)
            (sobremesa A) (libre B)
            (sobremesa B) (manovacia))

;; Objetivos (Desires)
(:goal (and (sobre A C) (sobre C B)))
```





# Agentes de razonamiento práctico

- Planificación
  - a. Lenguaje de representación
    - i. **De planes.**
  - b. Algoritmos

## domain.pddl

```
(:types Block - Object)
(:action Desapilar
:parameters (?X ?Y - Block)
:condition (and (manovacia) (libre ?X)
               (sobre ?X ?Y))
:duration (= ?duration 25)
:effects (and (not (manovacia))
              (not (libre ?X))
              (not (sobre ?X ?Y))
              (cogido ?x) (libre ?Y)))
```

## problem.pddl

```
(:objects A, B, C - Block)
(:init (and (libre C) (sobre C A)
            (sobremesa A) (libre B)
            (sobremesa B) (manovacia)
            (:goal (and (sobre A C) (sobre (C B)))))
```

## PDDL Planner

FF  
LPG

1. Llamar al planner
2. Capturar el plan y empezar a ejecutarlo desde el agente

```
MiAgente.java x
package agente2;

import es.upv.dsic.gti_ia.core.AgentID;
import es.upv.dsic.gti_ia.core.SingleAgent;

public class MiAgente extends SingleAgent {

    public MiAgente(AgentID aid) {
        super(aid);
    }

    // public void ini();

    // MiAgente agenteSmith = new MiAgente(new AgentID("Smith"));
    public void execute() {
        System.out.println("Hello world this is agent "+this.getName());
    }

    // public void finalize();
}
```

## plan.pddl

```
At 0: Desapilar(C,A)
At 25: Apilar(C,B)
At 50: Coger(A)
At 75: Apilar(A,C)
```

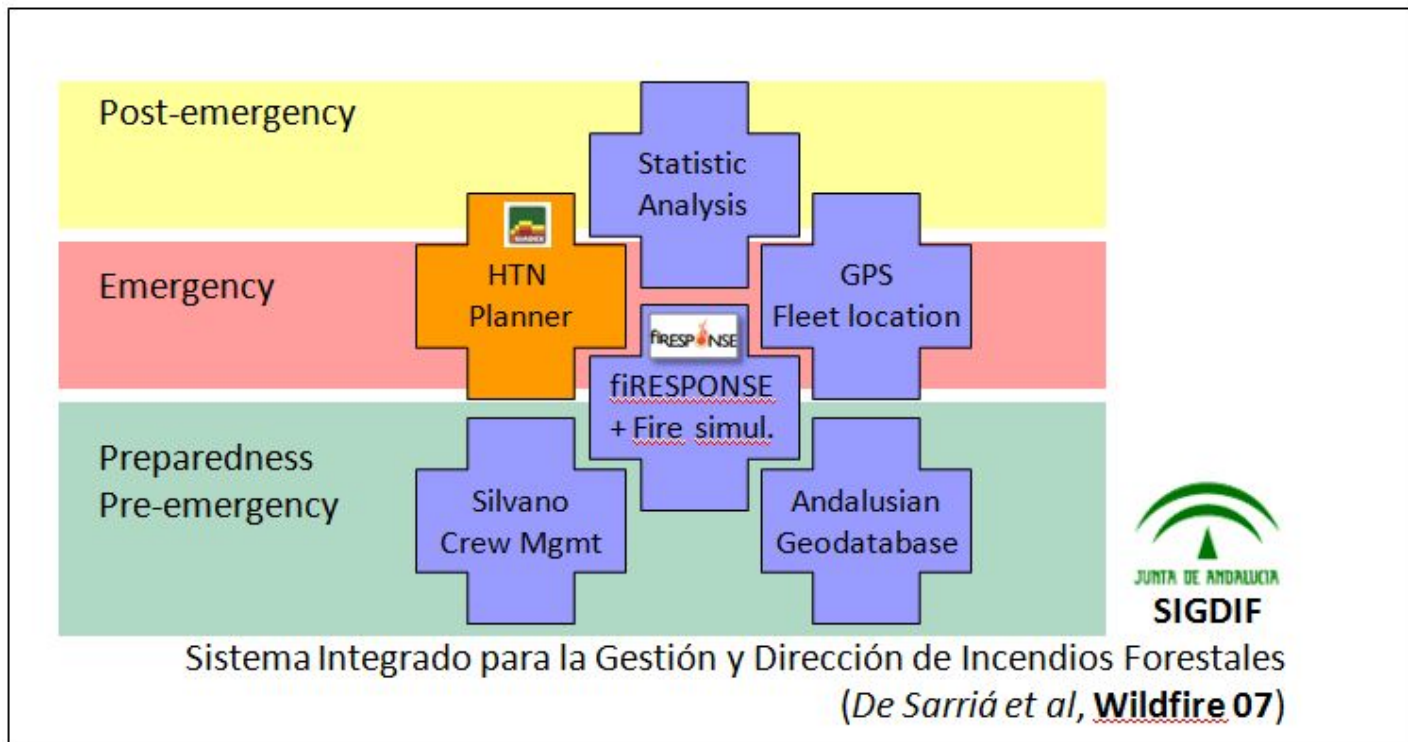
# Agentes de razonamiento práctico

Proyecto  
InfoSiadex

IActive  
Intelligent Technologies

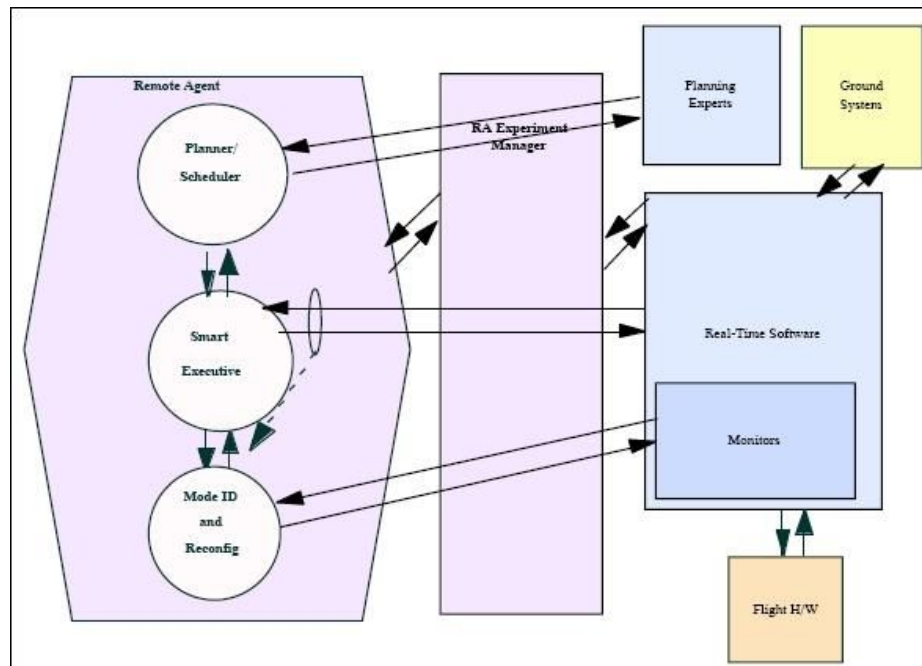
[+Info](#)

- IActive Decisor (planificación jerárquica)
  - Agente Reactivo/Deliberativo con aprendizaje
  - Silent Mode



# Agentes de razonamiento práctico

- NASA Remote Agent Experiment (RAX'99)
  - Misión “Deep Space 1”
  - Un agente planificador que controló durante 3 días una sonda espacial
  - Recibió el conocimiento ya en órbita



# Agentes de razonamiento práctico

## Algunos problemas

- **Inherentemente compleja.** Factor diferenciador de la inteligencia humana (y algunos primates)
  - Predicción del futuro
  - Uso de herramientas
  - Definición de procesos



# Agentes de razonamiento práctico

## Algunos problemas

- **Computacional.** No es un problema sencillo, requiere tiempo (exponencial)
  - El mundo puede cambiar mientras responde
- **Interferencias.** El plan de un agente no debe interferir con otros planes (de él o de otros) → **Proyecto comorbilidad**
- **Goal-oriented.** Un agente no abandona sus objetivos si un plan falla → prueba otro plan para los mismos objetivos → **¿se puede abandonar un objetivo?**

# Agentes de razonamiento práctico

## Algunos problemas

- **Razonamiento sobre objetivos.** Un agente siempre cree que sus objetivos son alcanzables
- **Certeza.** Un agente nunca intenta un plan si sabe que no funcionará
  - InfoSiadex → cambios de turno en retenes
- **Eficacia.** Si un agente tiene un plan es porque cree que tendrá éxito con el plan.

# Agentes de razonamiento práctico

## Algunos problemas

- **Efectos laterales.** Un agente no planifica los efectos laterales de un plan, pero algunos efectos laterales pueden ser beneficiosos para un plan.
- **Compleitud.** Se espera que el estado inicial sea completo

# Agentes de razonamiento práctico

## Algunas ventajas

- **Conocimiento transferible.** De un agente a otro “sin más” que transferir la parte correspondiente de PDDL
- **Real world.** Algunos lenguajes son usados en aplicaciones reales muy poderosas → capacidad práctica del agente **muy potente**