

# Agentes inteligentes e IA

## Inteligencia Artificial - 2018

Prof. Dr. Ariel Monteserín

amontese@exa.unicen.edu.ar - <http://www.exa.unicen.edu.ar/~amontese/>

ISISTAN – Fac. de Ciencias Exactas- UNICEN

CONICET

## Agenda

- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitectura abstracta
    - Arquitectura BDI
    - Reinforcement learning
- Sistemas multiagentes
  - Conceptos
  - Comunicación
  - Coordinación
    - Planning
    - Negociación



## Inteligencia artificial distribuida

- Sistemas inteligentes compuestos por un conjunto de agentes.
  - Inteligencia artificial + Sistemas distribuidos

## Agentes

- Agentes en la vida real



## Agentes: definiciones

- Rusell & Norving

*“Un agente es cualquier cosa que pueda ver en su **entorno** a través de **sensores** y actuar en su entorno a través de **efectores**”.*

- Nwana

*“Agente: componente de software y/o hardware el cual es capaz de actuar de manera precisa con el fin de realizar tareas en **representación** de su usuario”.*



## Agentes: definiciones...

- Shoham

*“Un agente es una entidad cuyo estado es visto como compuesto de **componentes mentales** tales como creencias, elecciones, aptitudes y compromisos”.*

- Maes

*“Un agente **autónomo** es un sistema computacional que habita en un entorno dinámico-complejo, **percibiendo** y **actuando** autónomamente en este **entorno**, y haciendo esto para realizar un conjunto de **objetivos** o tareas para los cuales fueron diseñados”.*



## Agentes: más definiciones...

- Wooldridge & Jennings

*“Un agente es un **sistema computacional** que está situado en algún **ambiente**, y que es capaz de actuar **autónomamente** en dicho ambiente con el fin de cumplir sus **objetivos**”.*



## Agentes: más definiciones...

- FIPA (Foundation for Intelligent Physical Agents)

*“Un agente es una entidad de software encapsulado con su propio **estado**, **conducta**, **hilo de control** y la **habilidad** para interactuar y **comunicarse** con otras entidades (gente, otros agentes o sistemas)”.*

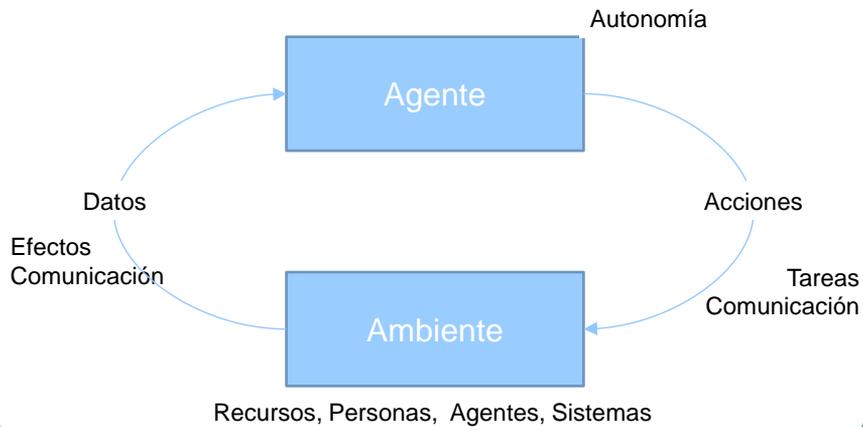


## Agenda

- Agentes
  - Definición
  - Agentes inteligentes
  - **Arquitectura abstracta**
    - Arquitectura BDI
    - Reinforcement learning
- Sistemas multiagentes
  - Conceptos
  - Comunicación
  - Coordinación
    - Planning
    - Negociación



## Vista abstracta de un agente



## Ejemplos de agentes

- Sistemas de control

- Termostato



- Alarma



- Software daemons

## Agentes inteligentes

- Wooldridge & Jennings

“Un **agente inteligente** es un agente capaz de ejecutar **flexiblemente acciones autónomas** con el fin de **cumplir sus objetivos**”, donde flexibilidad significa:

- **Reactividad**: capacidad de percibir su ambiente, y responder sin demoras a cambios que ocurren en él.
- **Pro-actividad**: capacidad de exhibir un comportamiento dirigido a objetivos, tomando la iniciativa.
- **Habilidad social**: capacidad de interactuar con otros agentes (y posiblemente humanos) a través de un lenguaje de comunicación.

## Agentes inteligentes

- |                    |              |
|--------------------|--------------|
| ● Características  | ● Termostato |
| ● Autonomía        | Si           |
| ● Reactividad      | Si           |
| ● Pro-actividad    | No           |
| ● Habilidad social | No           |

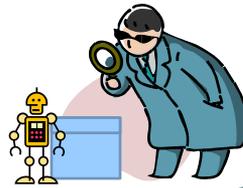


## Otras cuestiones...

- Adaptación
  - Implica percibir el entorno y reconfigurar el estado mental en respuesta a este entorno.
- Aprendizaje
  - Mediante distintas técnicas de aprendizaje de máquina.

## Agentes y objetos

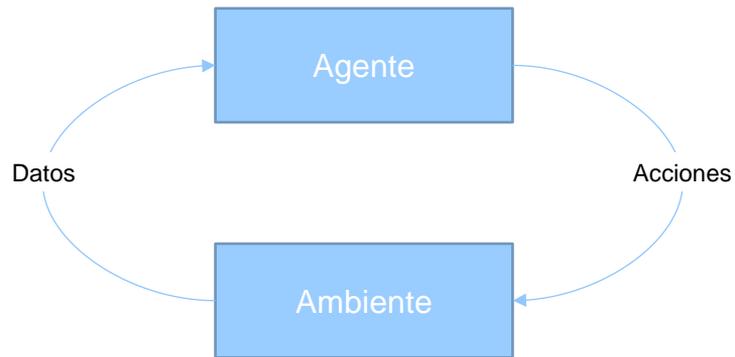
- **Objetos:**  
Entidades computacionales que *encapsulan* algún estado, son capaces de realizar acciones, o *métodos* en este estado, y se comunican mediante el *envío de mensajes*.
- **Agentes vs. Objetos**
  - Autonomía
  - Conducta flexible
  - Control



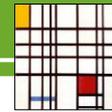
## Agentes y sistemas expertos

- **Sistema experto**  
Sistema capaz de resolver problemas o dar consejos en algún dominio rico en conocimiento.
- **Agentes vs. SE**
  - SE no interactúan directamente con el ambiente.

## Vista abstracta de un agente



## Arquitectura abstracta de un agente inteligente



- Ambiente  
 $S = \{s_1, s_2, \dots\}$
- Acciones  
 $A = \{a_1, a_2, \dots\}$
- Historia
- Agente estándar  
 $action: S^* \rightarrow A$
- Ambiente no determinista  
 $env: S \times A \rightarrow \wp(S)$

$$h: s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots \xrightarrow{a_{u-1}} s_u \xrightarrow{a_u} \dots$$



## Agentes puramente reactivos

- Toma de decisiones enteramente basada en el presente.

$$\text{action}: S \rightarrow A$$

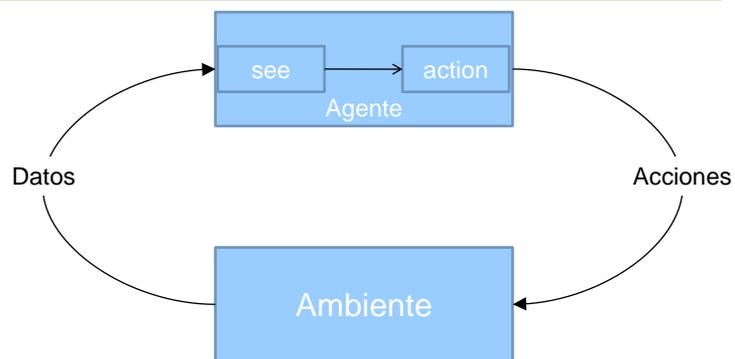
- Termostato

- $S = \{s \geq \text{temperatura}, s < \text{temperatura}\}$

- $A = \{\text{calefacción\_on}, \text{calefacción\_off}\}$

$$\text{action}(s) = \begin{cases} \text{calefacción off} & \text{si } s \geq \text{temperatura} \\ \text{calefacción on} & \text{cualquier otra} \end{cases}$$

## Percepción y acción



$$see: S \rightarrow P$$

$$action: P^* \rightarrow A$$

## Percepción y acción

### ● Ventajas de la división

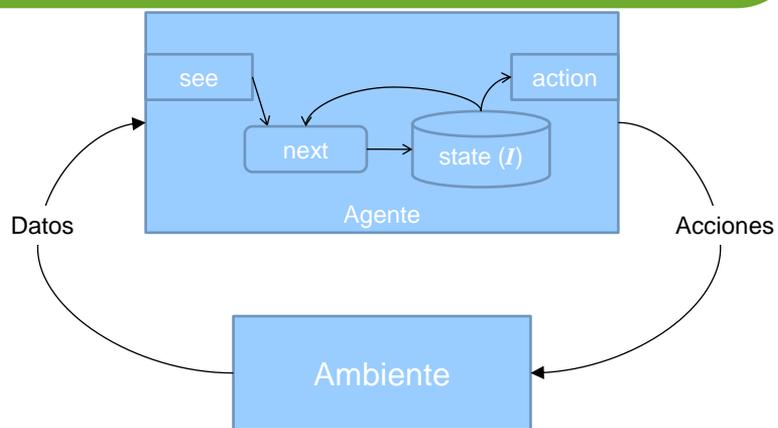
#### ● Sea

- X: “La temperatura de la habitación es Ok”.
- Y: “Ventana cerrada”.

$$S = \{\{-x, -y\}, \{-x, y\}, \{x, -y\}, \{x, y\}\}$$

$$see(s) = \begin{cases} p_1 & \text{si } s = s_1 \\ p_2 & \text{si } s = s_2 \\ p_3 & \text{si } s = s_3 \text{ o } s_4 \end{cases} \quad action(p) = \begin{cases} cerrarvent & p_1 \\ calefacción\ on & p_2 \\ calefacción\ off & p_3 \end{cases}$$

## Agentes con estado



$$see : S \rightarrow P \quad action : I \rightarrow A \quad next : I \times P \rightarrow I$$

## Agenda

- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitectura abstracta
    - **Arquitectura BDI**
    - Reinforcement learning
- Sistemas multiagentes
  - Conceptos
  - Comunicación
  - Coordinación
    - Planning
    - Negociación



## Arquitectura BDI

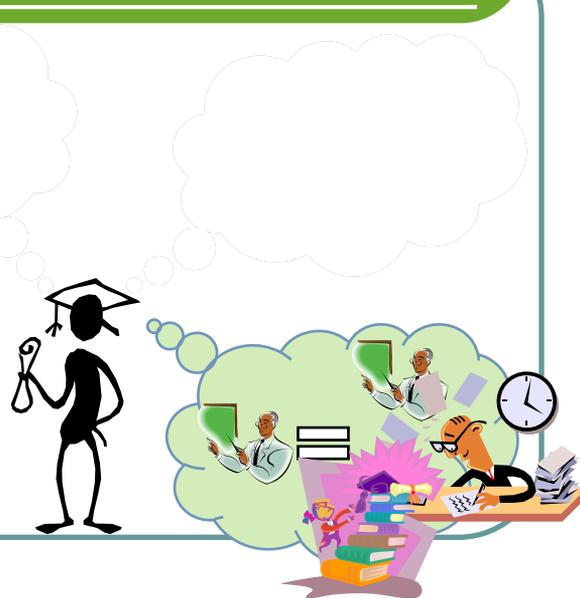
- **Belief-Desire-Intention**
- Razonamiento práctico
  - Deliberación
    - ¿Qué objetivos cumplir?
  - Razonamiento orientado a objetivos
    - ¿Cómo se cumplirán los objetivos?



## Ejemplo de razonamiento práctico



## Ejemplo de razonamiento práctico



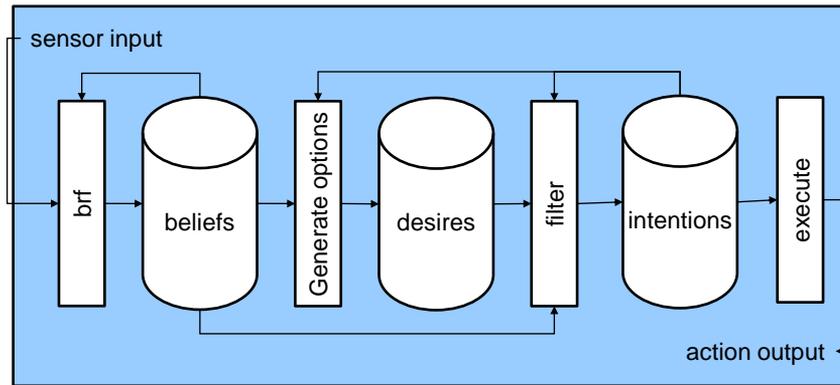
## Intenciones

- Rol de las intenciones en el razonamiento práctico
  - Dirigen el razonamiento basado en medios y fines (orientado a objetivos)
  - Restringen la deliberación futura
  - Persisten
  - Influencian las creencias bajo las cuales se basa el razonamiento práctico futuro

## Intenciones

- Reconsideración de intenciones
  - Un agente que no se detiene a reconsiderar sus intenciones (reactivo – orientado a eventos).
  - Un agente que constantemente reconsidera sus intenciones (proactivo – orientado a objetivos).
- Taza de cambio en el mundo  $\gamma$ 
  - Si  $\gamma$  es baja, los agentes que nunca se detienen a reconsiderar funcionan mejor.
  - Si  $\gamma$  es alta, los agentes que se detienen constantemente funcionan mejor.

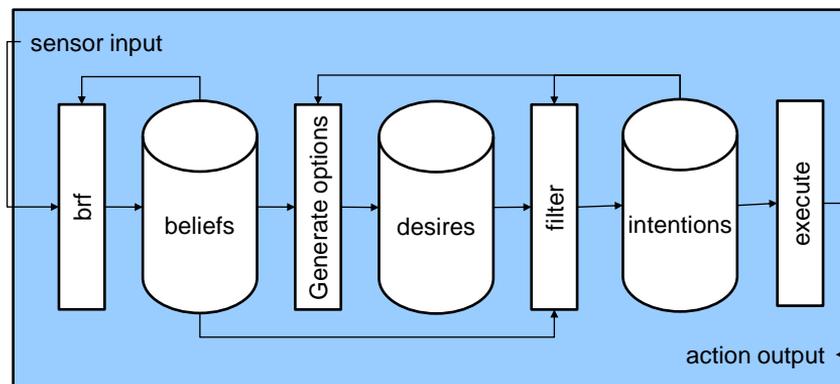
## Proceso de razonamiento práctico en agentes BDI



- Estado agente BDI

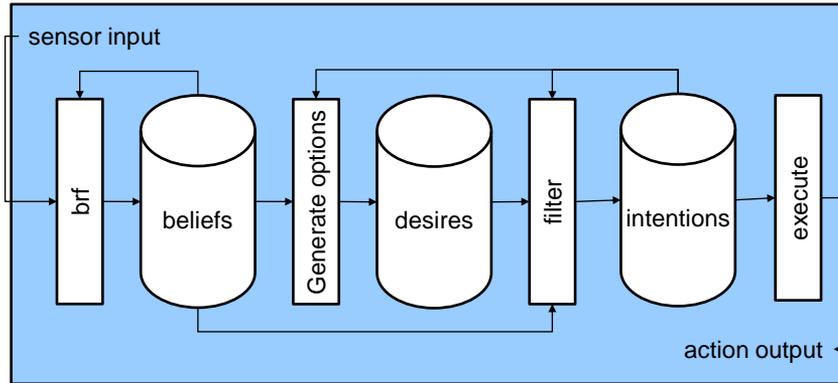
- $(B, D, I)$ , donde  $B \subseteq Bel$ ,  $D \subseteq Des$ ,  $I \subseteq Int$

## Proceso de razonamiento práctico en agentes BDI



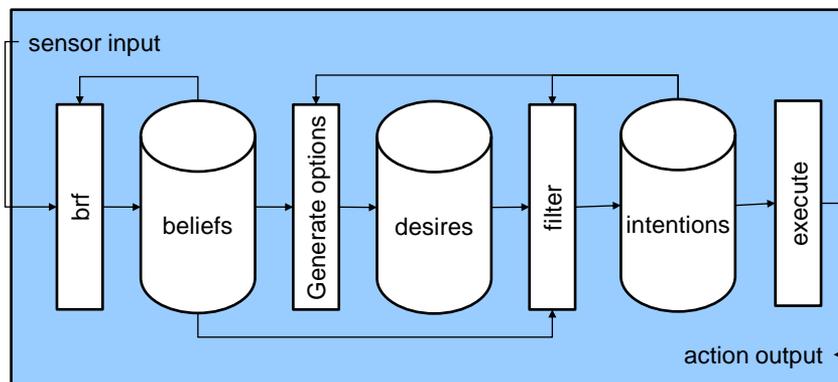
$$brf : \wp(Bel) \times P \rightarrow \wp(Bel)$$

## Proceso de razonamiento práctico en agentes BDI



$$options : \wp(Bel) \times \wp(Int) \rightarrow \wp(Des)$$

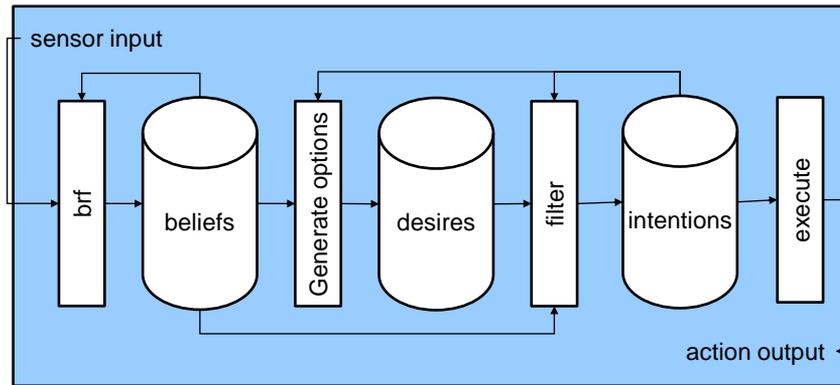
## Proceso de razonamiento práctico en agentes BDI



$$filter : \wp(Bel) \times \wp(Des) \times \wp(Int) \rightarrow \wp(Int)$$

$$\forall B \in \wp(Bel), \forall D \in \wp(Des), \forall I \in \wp(Int), filter(B, D, I) \subseteq I \cup D$$

## Proceso de razonamiento práctico en agentes BDI



$$execute : \wp(Int) \rightarrow A$$

$$action : P \rightarrow A$$

## Función BDI *action*

1. function  $action(p:P):A$
2. begin
3.      $B := brf(B, p)$
4.      $D := options(B, I)$
5.      $I := filter(B, D, I)$
6.     return  $execute(I)$
7. end function  $action$

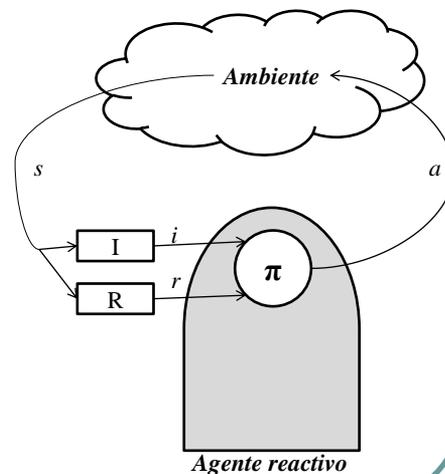
## Agenda

- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitectura abstracta
    - Arquitectura BDI
    - **Reinforcement learning**
- Sistemas multiagentes
  - Conceptos
  - Comunicación
  - Coordinación
    - Planning
    - Negociación



## Reinforcement learning

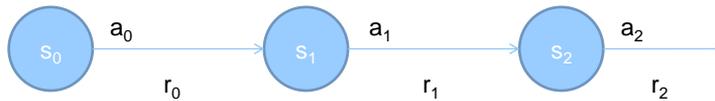
- Ambiente
  - $S = \{s_0, \dots, s_n\}$
- Set de acciones A
  - $A = \{a_0, \dots, a_m\}$
- Recompensa
  - $R: S \times A \rightarrow [0, 1]$
- Política
  - $\pi: I \times R \rightarrow A$



## Reinforcement learning

- **Objetivo**

- Aprender a seleccionar las acciones que maximizan la recompensa



$$r_0 + \gamma \cdot r_1 + \gamma^2 \cdot r_2 + \dots$$

- $\gamma$ : tasa de descuento ( $0 \leq \gamma \leq 1$ )

## Reinforcement learning

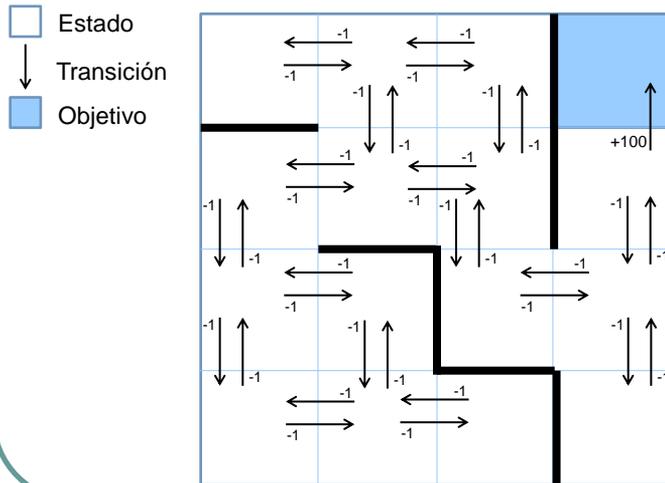
- **Formalización del problema**

$$r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i} = V_{\gamma}^{\pi}(s_t)$$

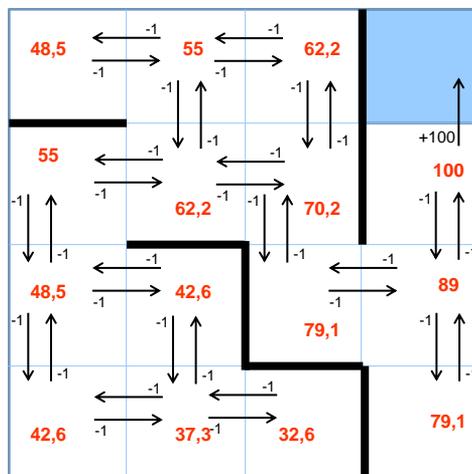
- ***Función estado-valor***

- $V_{\gamma}^{\pi}(s_t)$  *calcula las recompensas acumuladas que se consiguen siguiendo la política  $\pi$  a partir de un estado inicia  $s_t$*

# Reinforcement learning



# Reinforcement learning



Función estado-valor óptima

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

$$V^*([3,1]) = 100 = 100$$

$$V^*([3,2]) = -1 + \gamma \cdot 100 = 89$$

$$V^*([2,2]) = -1 + \gamma \cdot (-1) + \gamma^2 \cdot 100 = 79,1$$

$$V^*(s_t) = \max_{a \in A} [r(s_t, a) + \gamma \cdot V^*(\delta(s_t, a))]$$

$$\gamma = 0,9$$

## Reinforcement learning

- Funciones valor óptimas

- Función estado-valor óptima

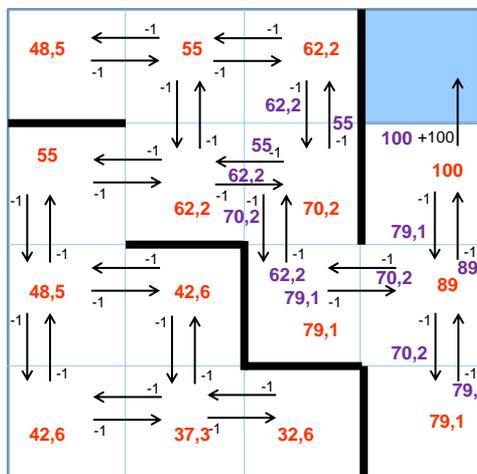
$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

- Función acción-valor óptima

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

$$Q^*(s, a) = r(s, a) + \gamma \cdot V^*(\delta(s, a))$$

## Reinforcement learning



Función acción-valor óptima

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

$$Q^*([3,1], \uparrow) = 100 = 100$$

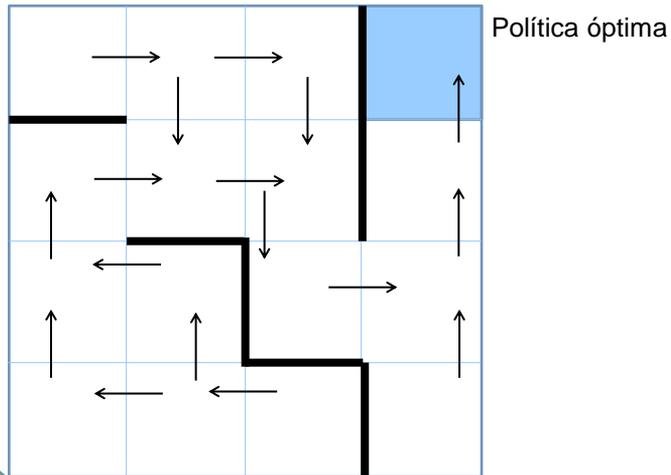
$$Q^*([3,2], \uparrow) = -1 + \gamma \cdot 100 = 89$$

$$Q^*([3,2], \leftarrow) = -1 + \gamma \cdot 79,1 = 70,2$$

$$Q^*(s, a) = r(s, a) + \gamma \cdot V^*(\delta(s, a))$$

$$\gamma = 0,9$$

## Reinforcement learning



## Reinforcement learning

```

for all  $(s,a)$  do
   $Q(s,a) \leftarrow$  inicializar
 $s \leftarrow$  estado observado
 $a \leftarrow$  seleccionar acción // Política
while  $s \neq$  estado final do
  Ejecutar  $a$ 
   $r \leftarrow$  recompensa
   $s' \leftarrow$  nuevo estado
   $a' \leftarrow$  nueva acción
   $Q(s,a) \leftarrow$  actualizar
   $s \leftarrow s'$ 
   $a \leftarrow a'$ 
end while

```

## Reinforcement learning

```

for all  $(s,a)$  do
     $Q(s,a) \leftarrow$  inicializar
 $s \leftarrow$  estado observado
 $a \leftarrow$  seleccionar acción // Política
while  $s \langle \rangle$  estado final do
    Ejecutar  $a$ 
     $r \leftarrow$  recompensa
     $s' \leftarrow$  nuevo estado
     $a' \leftarrow$  nueva acción
     $Q(s,a) \leftarrow r + \gamma \cdot \max_b Q(s', b)$ 
     $s \leftarrow s'$ 
     $a \leftarrow a'$ 
end while

```

Q-learning  
Ambientes Deterministas

## Reinforcement learning

```

for all  $(s,a)$  do
     $Q(s,a) \leftarrow$  inicializar
 $s \leftarrow$  estado observado
 $a \leftarrow$  seleccionar acción // Política
while  $s \langle \rangle$  estado final do
    Ejecutar  $a$ 
     $r \leftarrow$  recompensa
     $s' \leftarrow$  nuevo estado
     $a' \leftarrow$  nueva acción
     $Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \cdot \max_b Q(s', b) - Q(s,a)]$ 
     $s \leftarrow s'$ 
     $a \leftarrow a'$ 
end while

```

Q-learning  
Ambientes **NO** Deterministas  
 $0 < \alpha < 1$ : learning rate

## Reinforcement learning

- Fórmula

$$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \cdot \max_b Q(s', b) - Q(s,a)]$$

- Estimación vieja
- Nueva información
- Estimación vieja
- Estimación actualizada

## Reinforcement learning

- Q-learning garantiza la convergencia de  $Q(s,a)$  a  $Q^*(s,a)$  si

- El ambiente se puede modelar como un MDP
- En cada actualización  $i$  de  $Q(s,a)$  el  $\alpha$  es tal que
  - $\sum_{i=0}^{\infty} \alpha = \infty$  y  $\sum_{i=0}^{\infty} \alpha_i^2 < \infty \rightarrow \alpha = \frac{1}{n(s,a)}$
- Cada pareja  $(s,a)$  es visitada un número infinito de veces.
- El entorno es estacionario (la probabilidad de transitar de  $s$  a  $s'$  ejecutando  $a$  no varía)

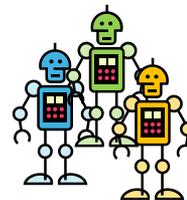
## Agenda

- **Agentes**
  - Definición
  - Agentes inteligentes
  - Arquitectura abstracta
    - Arquitectura BDI
    - Reinforcement learning
  
- **Sistemas multiagentes**
  - Conceptos
  - Comunicación
  - Coordinación
    - Planning
    - Negociación

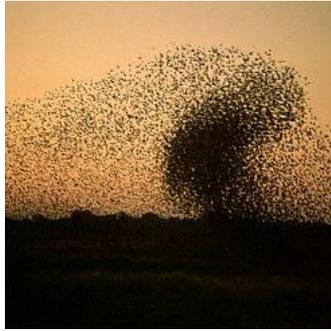


## Sistemas multiagentes

- Un sistema multiagente es un sistema compuesto de múltiples agentes inteligentes, capaces de cumplir objetivos que son difíciles de alcanzar por un sistema individual.

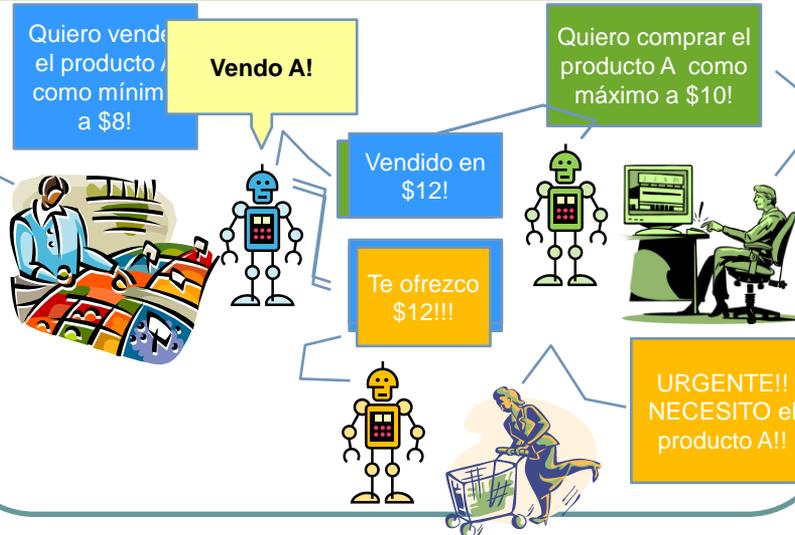


## Ejemplo

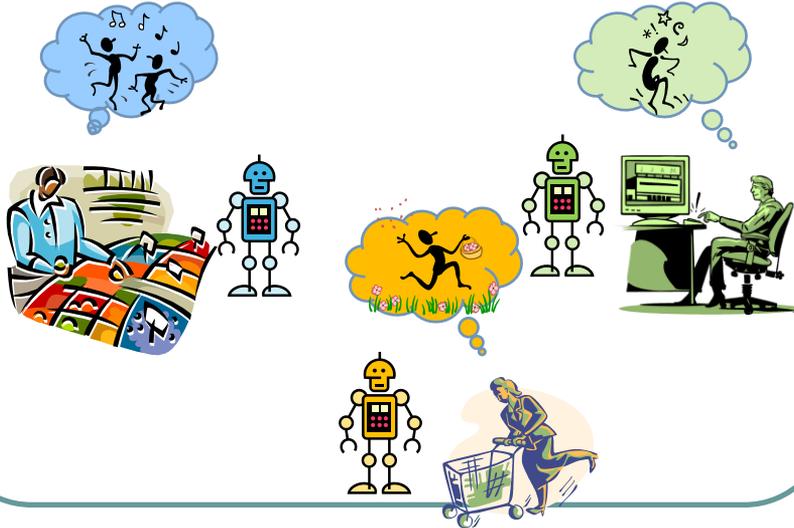


- No controladores
- No organizadores
- Auto-organización
- Reglas simples
  - Mantener distancia
  - Ser sociables
- Comportamiento complejo

## Ejemplo

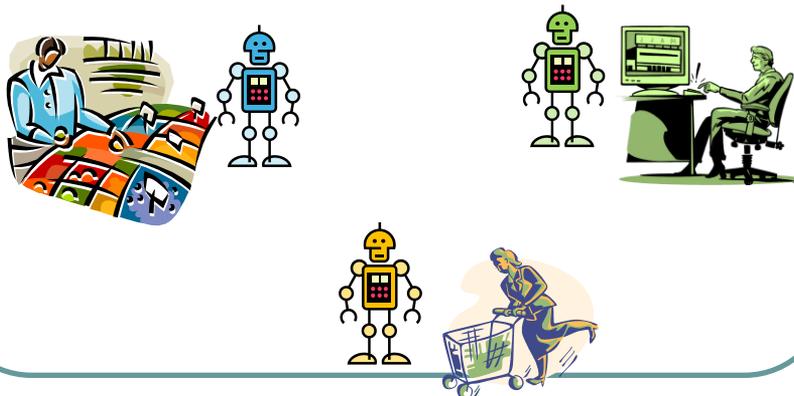


## Ejemplo

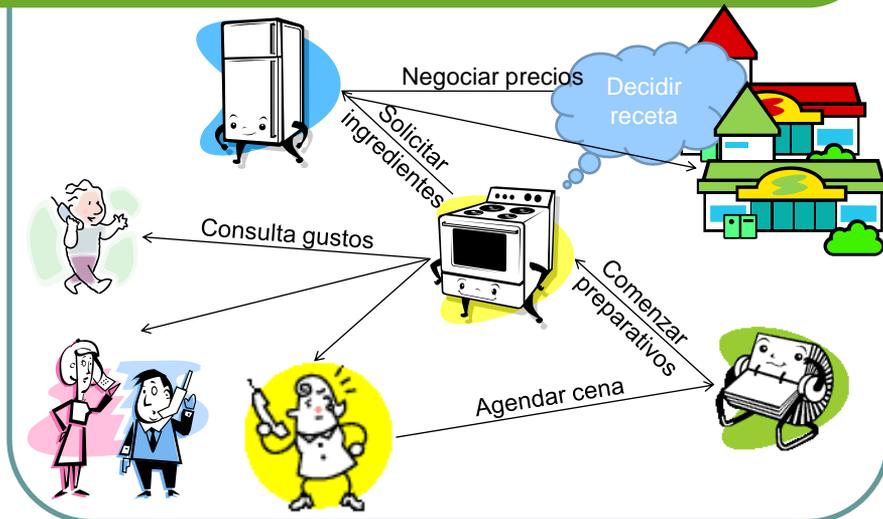


## Ejemplo

- Información privada
- Competitividad
- Confianza
- Reputación



## Ejemplo cooperativo



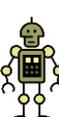
## PUMAS-GR



## PUMAS-GR




Mascotas  
Civil War  
Frente al Mar  
Forest Gump  
Megamente

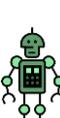







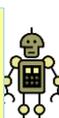


## PUMAS-GR

Mascotas	5
Megamente	4,5
Civil War	4
Forest Gump	2
Frente al Mar	1

Forest Gump	5
Frente al Mar	4
Megamente	4
Mascotas	2
Civil War	1

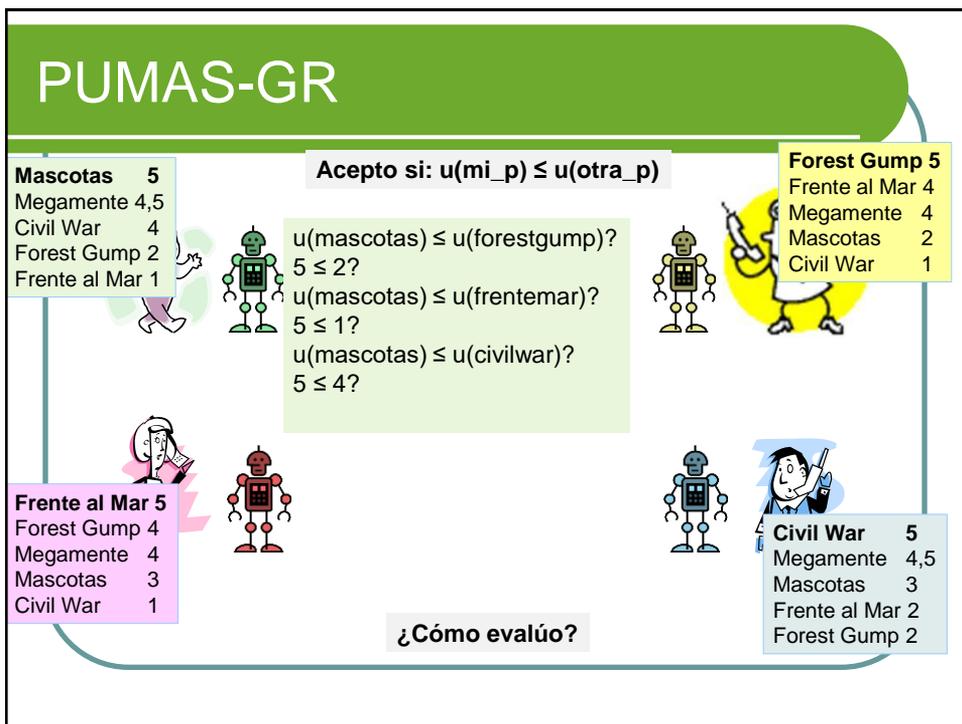
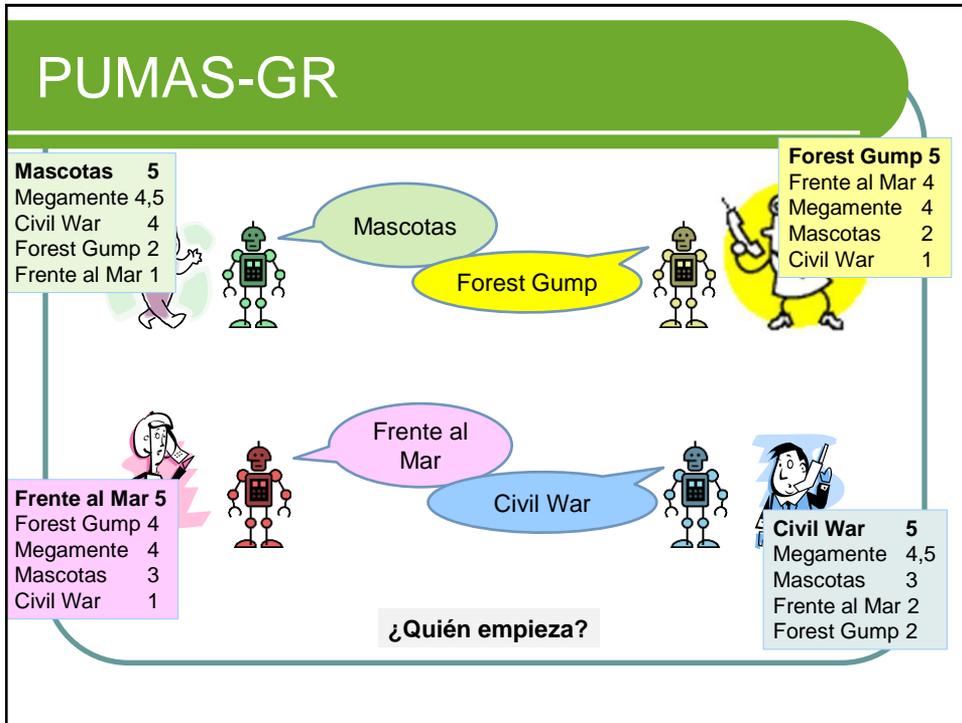






Frente al Mar	5
Forest Gump	4
Megamente	4
Mascotas	3
Civil War	1

Civil War	5
Megamente	4,5
Mascotas	3
Frente al Mar	2
Forest Gump	2



## PUMAS-GR

<b>Mascotas</b>	5
Megamente	4,5
Civil War	4
Forest Gump	2
Frente al Mar	1

<b>Forest Gump</b>	5
Frente al Mar	4
Megamente	4
Mascotas	2
Civil War	1

NO! NO!

<b>Frente al Mar</b>	5
Forest Gump	4
Megamente	4
Mascotas	3
Civil War	1

<b>Civil War</b>	5
Megamente	4,5
Mascotas	3
Frente al Mar	2
Forest Gump	2

NO! NO!

## PUMAS-GR

<b>Mascotas</b>	5
Megamente	4,5
Civil War	4
Forest Gump	2
Frente al Mar	1

<b>Forest Gump</b>	5
Frente al Mar	4
Megamente	4
Mascotas	2
Civil War	1

**¡Alguien debe conceder!  
¿Quién?**

$Z = (5-1)/5 = 0,8$

$Z = (5-1)/5 = 0,8$

$$Z_i = \begin{cases} 1 & \text{si } u_i(x_i) = 0 \\ \frac{u_i(x_i) - \min\{u_i(x_k) | k \in A\}}{u_i(x_i)} \end{cases}$$

$Z = (5-1)/5 = 0,8$

$Z = (5-2)/5 = 0,6$

**El agente con menor valor Zeuthen concede.**

<b>Frente al Mar</b>	5
Forest Gump	4
Megamente	4
Mascotas	3
Civil War	1

<b>Civil War</b>	5
Megamente	4,5
Mascotas	3
Frente al Mar	2
Forest Gump	2

## PUMAS-GR

<b>Mascotas</b>	<b>5</b>
Megamente	4,5
Civil War	4
Forest Gump	2
Frente al Mar	1

**¿Cómo conceder?**

$u(\text{megamente}) \leq u(\text{megamente})?$   
 $4.5 \leq 4.5?$

$u(\text{frentemar}) \leq u(\text{megamente})?$   
 $4 \leq 4?$

<b>Forest Gump</b>	<b>5</b>
Frente al Mar	4
Megamente	4
Mascotas	2
Civil War	1

<b>Frente al Mar</b>	<b>5</b>
Forest Gump	4
Megamente	4
Mascotas	3
Civil War	1

**Megamente?**

$u(\text{forestgump}) \leq u(\text{megamente})?$   
 $4 \leq 4?$

<b>Civil War</b>	<b>5</b>
<b>Megamente</b>	<b>4,5</b>
Mascotas	3
Frente al Mar	2
Forest Gump	2

**Evaluación alternativa:**  
Acepto si:  $u(\text{prox}_p) \leq u(\text{otra}_p)$

## Swarm intelligence

- Técnica de IA basada en el estudio del comportamiento colectivo en sistemas descentralizados y autoorganizativos.
- La inteligencia colectiva emergente de un grupo de agentes simples.
- Algoritmos o mecanismos distribuidos de resolución de problemas inspirados en el comportamiento colectivo de colonias de insectos sociales u otras sociedades de animales.

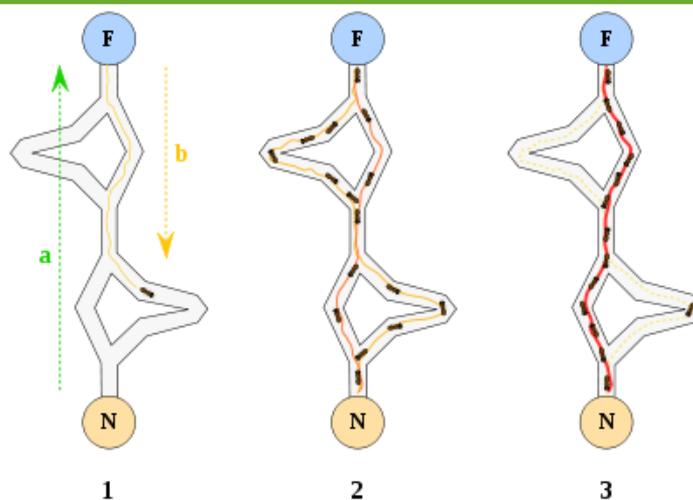


## Swarm intelligence

- Algoritmos
  - Colonia de hormigas (ACO)
  - Optimización por enjambre de partículas
  - Colonia de abejas
  - Sistemas inmunes artificiales



## Ant Colony Optimization



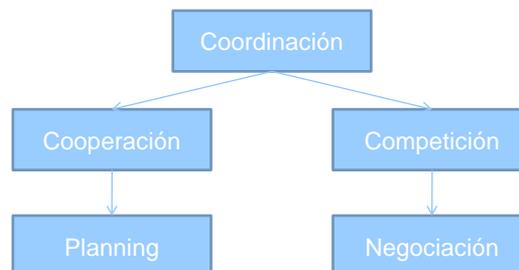
## Agenda

- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitectura abstracta
    - Arquitectura BDI
    - Reinforcement learning
- Sistemas multiagentes
  - Conceptos
  - **Comunicación**
  - Coordinación
    - Planning
    - Negociación



## Comunicación entre agentes

- Percepción
  - Recibir mensajes
- Acción
  - Enviar mensajes

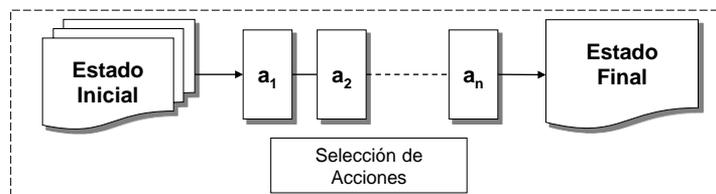


## Agenda

- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitectura abstracta
    - Arquitectura BDI
    - Reinforcement learning
- Sistemas multiagentes
  - Conceptos
  - Comunicación
  - Coordinación
    - **Planning**
    - Negociación



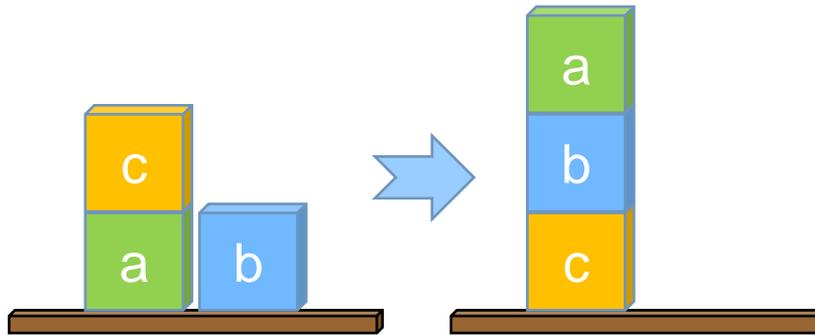
## Planning



- Problema de planning definido por la tripla  $\langle i, f, A \rangle$ .
  - Estado Inicial  $i$ : descripción del mundo.
  - Estado Final  $f$ : objetivo que se quiere alcanzar.
  - Acciones disponibles para la construcción del plan ( $A$ ) las cuales poseen precondiciones y efectos.
- Mecanismo de selección de acciones.

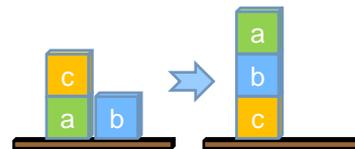


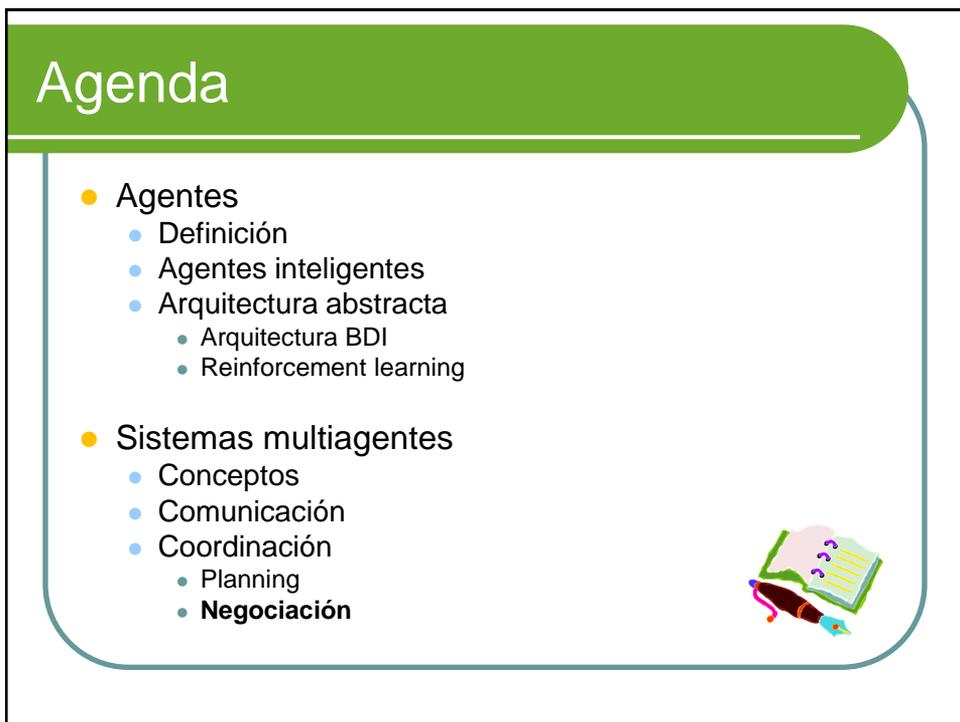
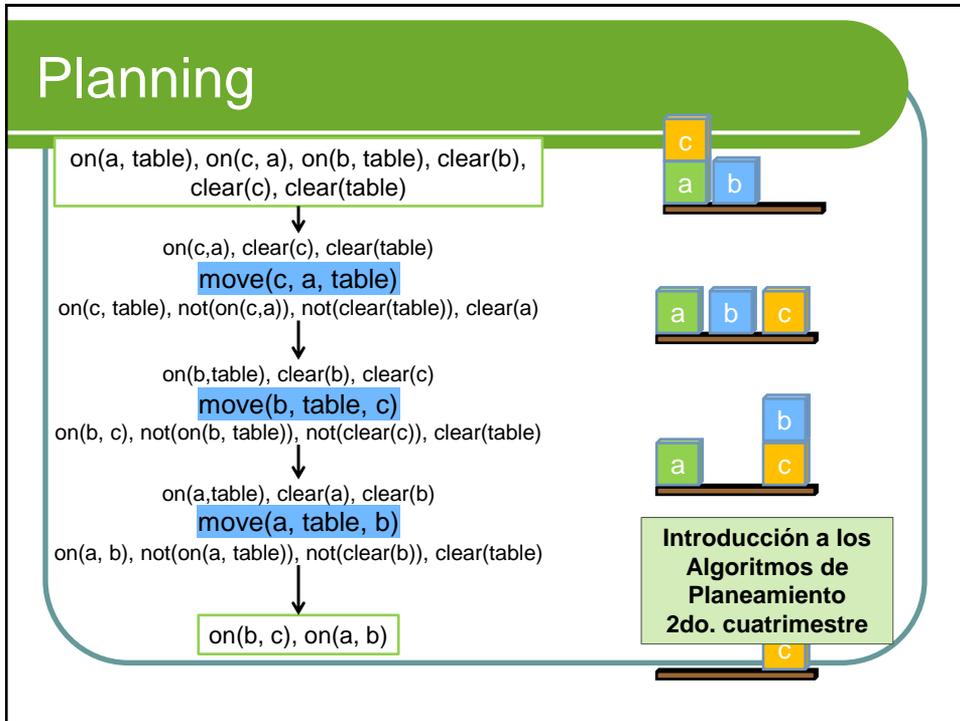
## Planning



## Planning

- Estado inicial
  - on(a, table), on(c, a), on(b, table), clear(b), clear(c), clear(table)
- Estado final
  - on(b, c), on(a, b)
- move(X, Y, Z) // Mover X de Y a Z
  - Pre: on(X,Y), clear(X), clear(Z)
  - Post: on(X, Z), not(on(X,Y)), not(clear(Z)), clear(Y)





## Agentes inteligentes

- Características principales:
  - Autonomía
  - Capacidad de aprendizaje
  - Comportamiento reactivo
  - Comportamiento proactivo
  - Habilidad social
    - Negociación

## Negociación: definición

- *La negociación es una forma de **interacción** en la cual un **grupo de agentes**, con **intereses conflictivos** y un **deseo de cooperar**, intentan alcanzar un **acuerdo mutuamente aceptable** en la **división de recursos escasos**. [Rahwan 2004]*



## Protocolo de negociación

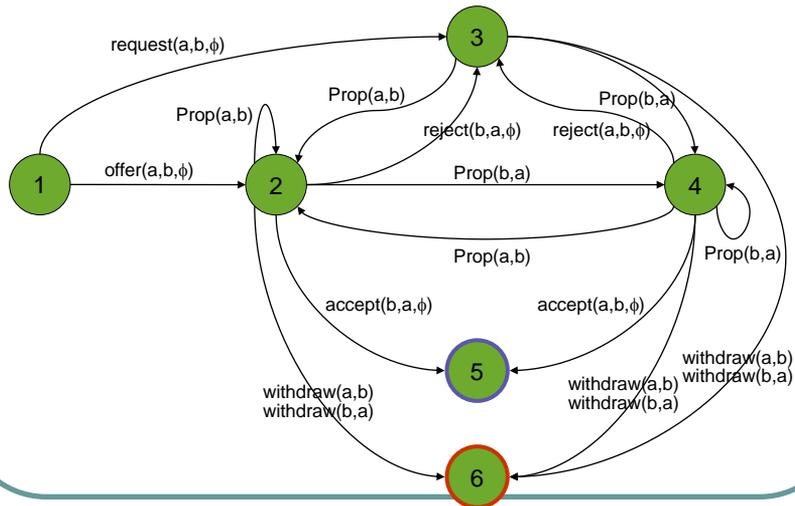
- Protocolos de negociación
  - Set formal de convenciones que gobiernan la interacción entre los participantes.
  - Restringe el uso de los lenguajes de comunicación y dominio.
    - Protocolo de interacción
    - Otras reglas

## Protocolos de negociación

- Protocolo de interacción
  - Especifica para cada etapa del proceso de negociación qué está permitido decir.
    - Máquina de estados finitos
    - Juegos de diálogos

## Protocolo de interacción

### Ejemplo: Máquina de estados finitos



## Clasificación de modelos de negociación

- Según técnica de negociación utilizada.
- Según la cantidad de agentes negociando.
- Según la cantidad de temas/items negociados.
- Según la privacidad de la información.
- Según la relación entre los agentes.

## Según técnica de negociación

- Según la técnica de negociación utilizada:
  - Basados en Teoría de juego
  - Basados en Argumentación

## Teoría de juegos

- Se utilizan las técnicas de teoría de juegos para estructurar y organizar la negociación.
- Determinación de estrategias óptimas. Dependiendo de:
  - Objetivos del agente.
  - Precios mínimos y máximos.
  - Limite temporal (Deadline).
  - Información del oponente.
- Búsqueda del equilibrio.
- Intercambio de propuestas y contrapropuestas.

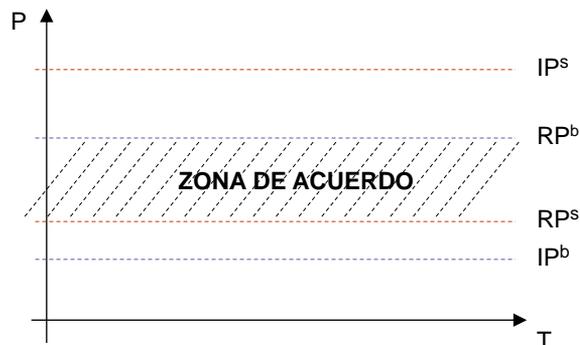


## Componentes del modelo de negociación

- Protocolo de negociación
- Estrategias de negociación
- Los estados de la información del agente

## El protocolo de negociación

- Agentes b (comprador) y s (vendedor)
- $[IP^a, RP^a]$  para  $a \in [b, s]$



## Protocolo de negociación

- Deadlines  $t^a \in T$



## Protocolo de negociación

- Ofertas:

- $p_{a \rightarrow \bar{a}}^t$
- Inicio aleatorio
- Función de utilidad  $U^a$  para evaluar ofertas.
- Acción del agente:

$$\bullet A^s(t, p_{b \rightarrow s}^t) = \begin{cases} \text{Quit} & \text{if } t > T^s \\ \text{Accept} & \text{if } U^s(p_{b \rightarrow s}^t) \geq U^s(p_{s \rightarrow b}^t) \\ \text{Offer } p_{s \rightarrow b}^t \text{ en } t' & \text{sino, para } t' = t + 1 \end{cases}$$

## Protocolo de negociación

- Función de utilidad
  - Función de utilidad de von Neumann – Morgenstern
    - $U^a(p,t) = U_p^a(p) \cdot U_t^a(t)$
  - La utilidad ante el conflicto es la más baja.

## Protocolo de negociación

- Función de utilidad en base al precio:

$$U^a(p) = \begin{cases} RP^b - p & \text{para el comprador b} \\ P - RP^s & \text{para el vendedor s} \end{cases}$$



## Protocolo de negociación

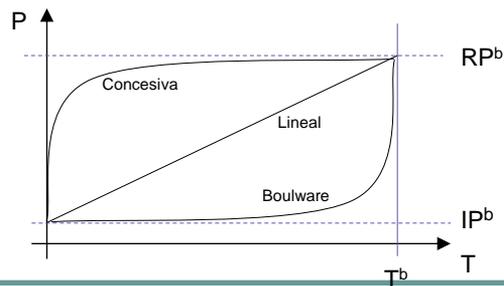
- Función de utilidad en base al tiempo
  - $U^a(t) = (\delta^a)^t$
  - $\delta^a$  es el factor de descuento de  $a$ . Así:
    - Cuando  $\delta^a > 1 \rightarrow a$  es paciente.
    - Cuando  $\delta^a < 1 \rightarrow a$  es impaciente.

## Generación de contraofertas

- Se asume dependencia del tiempo.
  - $p_{a \rightarrow \hat{a}}^t = \begin{cases} IP^a + \phi^a(t) \cdot (RP^a - IP^a) & \text{para } a = b \\ RP^a + (1 - \phi^a(t)) \cdot (IP^a - RP^a) & \text{para } a = s \end{cases}$
  - $\phi^a(t) = k^a + (1 - k^a) \cdot (t / T^a)^{1/\psi}$ 
    - $k^a \in [0, 1]$ , con  $k = 0$ 
      - $p_{a \rightarrow \hat{a}}^0$  sea igual a  $IP^a$
      - $p_{a \rightarrow \hat{a}}^T$  sea igual a  $RP^a$

## Generación de contraofertas

- Tres tácticas dependiendo del valor de  $\psi$ 
  - Boulware:  $\psi < 1$
  - Lineal:  $\psi = 1$
  - Concesiva:  $\psi > 1$



## Generación de contraofertas

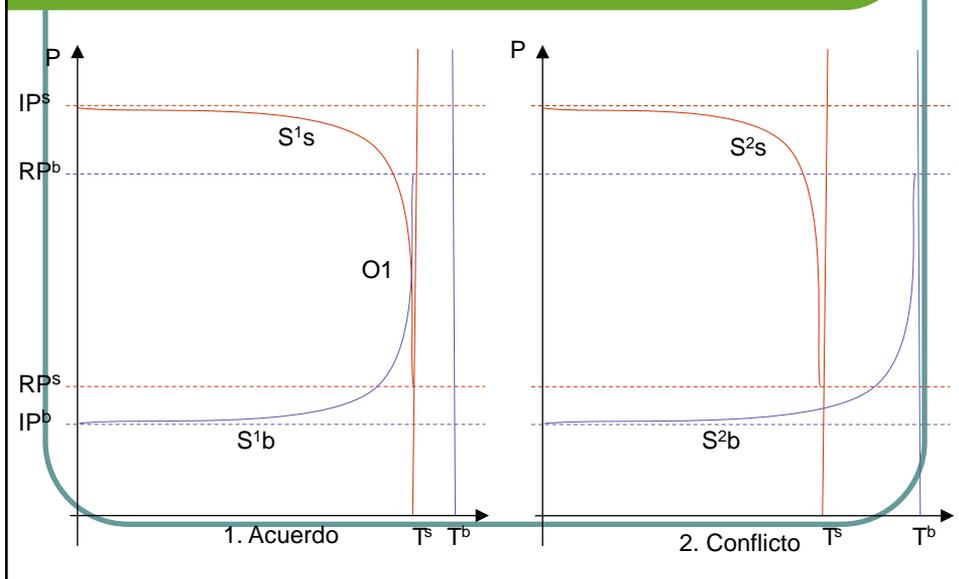
- Estrategias de generación
  - $S^a = (IP^a, FP^a, t^a, \psi^a)$ 
    - $IP^a$ : precio inicial.
    - $FP^a$ : precio mas allá del cual no se concederá.
    - $t^a$ : tiempo de  $FP^a$ .
    - $\psi^a$ : parametro de la función  $\phi^a(t)$ .

## Generación de contraofertas

- Resultado de la negociación
  - $O \in [(p, t), C]$ 
    - $(p, t)$  indica el precio y el tiempo del acuerdo.
      - $p \in [RP^s, RP^b]$
      - $t \in [0, \min(T^b, T^s)]$
    - $C$  representa el conflicto.



## Generación de contraofertas: ejemplos



## Estados de información

- Estrategia óptima → utilidad máxima
- Depende de la información que el agente tiene sobre los parámetros de negociación.
- Estado de información de  $a$ 
  - $I^a = (RP^a, T^a, U^a, S^a, L^{\hat{a}}_p, L^{\hat{a}}_t)$ 
    - $(RP^a, T^a, U^a, S^a)$  parámetros propios
    - $(L^{\hat{a}}_p, L^{\hat{a}}_t)$  creencias sobre el oponente
      - Distribuciones de probabilidad sobre el RP del oponente y su deadline.

## Estados de información

- $L^{\hat{a}}_t = ((T^{\hat{a}}_1, \alpha^{\hat{a}}_1), (T^{\hat{a}}_2, \alpha^{\hat{a}}_2), \dots, (T^{\hat{a}}_n, \alpha^{\hat{a}}_n))$ 
  - $T^{\hat{a}}_i$  posible deadline de  $\hat{a}$
  - $\alpha^{\hat{a}}_i$  probabilidad asociada a  $T^{\hat{a}}_i$
- $L^{\hat{a}}_p = ((RP^{\hat{a}}_0, \beta^{\hat{a}}_0), (RP^{\hat{a}}_1, \beta^{\hat{a}}_1), \dots, (RP^{\hat{a}}_m, \beta^{\hat{a}}_m))$ 
  - $RP^{\hat{a}}_i$  posible precio de reserva
  - $\beta^{\hat{a}}_i$  probabilidad asociada a  $RP^{\hat{a}}_i$

## Escenarios de negociación

- Seis escenarios en base a los deadlines y al factor de descuento  $\delta^a$

Escenarios	Rel. con deadline	Factor de desc.
N1	$T_n^{\hat{a}} < T^a$	$\delta^a > 1$
N2	$T_k^{\hat{a}} < T^a \leq T_{k+1}^{\hat{a}}$ para $k + 1 < n$	$\delta^a > 1$
N3	$T^a < T_1^{\hat{a}}$	$\delta^a > 1$
N4	$T_n^{\hat{a}} < T^a$	$\delta^a < 1$
N5	$T_k^{\hat{a}} < T^a \leq T_{k+1}^{\hat{a}}$ para $k + 1 < n$	$\delta^a < 1$
N6	$T^a < T_1^{\hat{a}}$	$\delta^a < 1$

## Escenarios de negociación

- Posibles escenarios en la interacción:

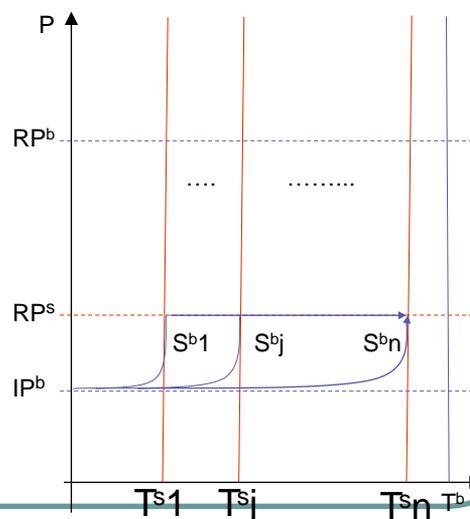
Agent $a$	Agent $\hat{a}$
N1	$N_2, N_3, N_5, N_6$
N2	$N_1, N_2, N_3, N_4, N_5, N_6$
N3	$N_1, N_2, N_4, N_5$
N4	$N_2, N_3, N_5, N_6$
N5	$N_1, N_2, N_3, N_4, N_5, N_6$
N6	$N_1, N_2, N_4, N_5$

## Estrategias óptimas

- Debe asegurar un acuerdo antes del  $T^a$  min.
- Para escenario  $N_1$ 
  - Si deadline  $s = T^s_i \rightarrow S^s_i$  con  $RP^s$  en  $T^s_i$
  - $\alpha^s_i$  es la prob. de  $T^s_i \rightarrow \alpha^s_i$  es la prob. de  $S^s_i$
  - $T^s_i < T^b \rightarrow$  maximizar  $U$  esperando  $RP^s$  y max  $T$
  - Entonces:
    - $\forall t < T^s_j \rightarrow S^b_j = (IP^b, RP^s, T^s_i, B) \rightarrow \min T = T^s_j$
    - $\forall T^s_j < t < T^s_n \rightarrow$  se ofrece  $RP^s \rightarrow \max T = T^s_n$
    - Si  $T^s < T^s_j \rightarrow \hat{c}$

## Estrategias óptimas

- Posibles estrategias para  $N_1$



## Estrategias óptimas

- $S_j^b$  que  $\text{Max}(EU^b_o) \rightarrow S^b_o$

$$EU^b_o = \sum_{x=1}^{j-1} \alpha^s_x U^b(C) + \alpha^s_j U^b(RP^s, T^s_j) + \sum_{y=j+1}^n \alpha^s_y U^b(RP^s, t)$$

donde  $T^s_j \leq t \leq T^s_n$

## Estrategias óptimas

Escenarios	t durante la neg.	$S^b_o$
N1	$t \leq T^s_j$	$(IP^b, RP^s, T^s_j, B)$
	$t > T^s_j$	$(RP^s, RP^s, T^s_n, L)$
N2	$t \leq T^s_j$	$(IP^b, RP^s, T^s_j, B)$
	$T^s_j \leq t \leq T^s_k$	$(RP^s, RP^s, T^s_k, L)$
	$t > T^s_k$	$(RP^s, RP^b, T^b, B)$
N3	$t \leq T^b$	$(IP^b, RP^b, T^b, B)$
N4	$t \leq T'$	$(IP^b, RP^s, T', C)$
	$t > T'$	$(RP^s, RP^s, T^s_n, L)$
N5	$t \leq T'$	$(IP^b, RP^s, T', C)$
	$T' \leq t \leq T^s_k$	$(RP^s, RP^s, T^s_k, L)$
	$t > T^s_k$	$(RP^s, RP^b, T^b, B)$
N6	$t \leq T'$	$(IP^b, RP^b, T', C)$
	$t > T'$	$(RP^b, RP^b, T^b, L)$

## Basados en teoría de juegos

- **Ventajas:**
  - Computacionalmente eficiente.
  - Adecuados para negociaciones bilaterales (vendedor – comprador, subastas).

## Basados en teoría de juegos

- **Desventajas**
  - Se asume que es posible distinguir las preferencias de los agentes con respecto a los resultados.
  - Dificultad para determinar la función de utilidad cuando los temas no son cuantificables.
  - Basados en la noción de racionalidad perfecta (Recursos ilimitados y espacio de resultados conocidos).

## Problemas en los anteriores modelos

- Solo intercambian ofertas y contraofertas.
- Se asume que el agente posee un mecanismo de evaluación de propuestas.
- Los agentes no pueden influir en el modelo de preferencias ni en las actitudes mentales de sus oponentes.

## Basados en argumentación

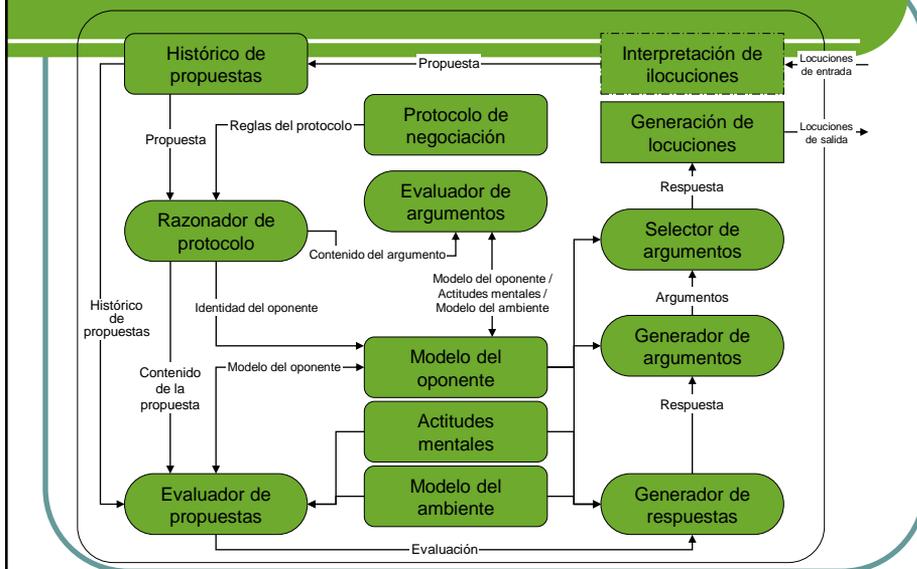
- Intercambiar información adicional.
- Argumentos:
  - Justificar una postura de negociación.
  - Influir la postura de negociación.
- Mejora el proceso de negociación.
- Relaciones de confianza y autoridad.

## Negociación basada en argumentación - Tendencias

- Existen dos grandes tendencias en la literatura sobre negociación basada en argumentación (Rahwan et al., 2005).
  - Enfoques para adaptar lógicas dialécticas para argumentación rebatible embebiendo conceptos de negociación dentro de ellas (Amgoud et al., 2000; Parsons et al., 1998; Rueda et al., 2002).
  - Enfoques para extender *frameworks* de negociación para que permitan a los agentes intercambiar argumentos retóricos, como recompensas y amenazas (Kraus et al., 1998; Ramchurn et al., 2003a).



## Arquitectura de un agente argumentador



## Basados en argumentación

- Argumentos persuasivos → Persuasión
- Tipo de argumentos retóricos:
  - Amenazas
  - Recompensas
  - Apelaciones

## Basados en argumentación

- Amenazas:
  - $threaten(a, \hat{a}, [not]\alpha_1, [not]\alpha_2, t_i)$ 
    - $a$ : agente emisor del argumento
    - $\hat{a}$ : agente receptor del argumento
    - $[not]\alpha_1$ : acción propuesta (causa)
    - $[not]\alpha_2$ : acción perjudicial (consecuencia)
    - $t_i$ : tiempo del argumento

## Basados en argumentación

- Recompensas:

- $reward(a, \hat{a}, [not]\alpha_1, [not]\alpha_2, t_i)$ 
  - $a$ : agente emisor del argumento
  - $\hat{a}$ : agente receptor del argumento
  - $[not]\alpha_1$ : acción propuesta (causa)
  - $[not]\alpha_2$ : acción beneficiosa (consecuencia)
  - $t_i$ : tiempo del argumento

## Basados en argumentación

- Apelaciones:

- $appeal(a, \hat{a}, [not]\beta_1, [not]\beta_2, t_i)$ 
  - $a$ : agente emisor del argumento
  - $\hat{a}$ : agente receptor del argumento
  - $[not]\beta_1$ : creencia o acción.
  - $[not]\beta_2$ : justificación.
  - $t_i$ : tiempo del argumento

## Basados en argumentación

- Ejemplo de amenazas:

- *threaten(a, b, not accept(b, a, time = 24h, t<sub>2</sub>),  
appeal(a, boss\_b, not b = apto,  
not accept(b, a, time = 24h, t<sub>2</sub>), t<sub>3</sub>), t<sub>1</sub>)*

## Basados en argumentación

- Ejemplo de recompensas:

- *reward(a, b, accept(b, a, time = 24hs, t<sub>2</sub>),  
pay(overtime, b, t<sub>3</sub>), t<sub>1</sub>)*

## Basados en argumentación

- Tipos de apelaciones:
  - De promesas anteriores
  - Contraejemplo
  - De practica predominante
  - De interés propio

## Basados en argumentación

- De promesas anteriores:
  - Se recuerda una recompensa anterior.
  - `appeal(b, a, pay(overtime,b,t5),  
reward(a, b, accept(b, a, time = 24hs, t2),  
pay(overtime,b,t3), t1) &&  
accept(b, a, time = 24hs, t2), t4)`

## Basados en argumentación

- Contraejemplos:

- `appeal(a, b, accept(b, a, time = 24hs, t2), accept(b, c, time = 24hs, t0), t3)`

## Basados en argumentación

- De práctica predominante:

- `appeal(a, b, accept(b, a, time = 24hs, t2), accept(d, a, time = 24hs, t0), t3)`

## Basados en argumentación

- De interés propio:
  - `appeal(a, b, accept(b, a, time = 24hs, t2), propose((c,d,e), b, new(contract), t3), t1)`

## Basados en argumentación

- Los agentes argumentadores son capaces de:
  - Generar argumentos de salida.
  - Seleccionar el mejor argumento.
  - Evaluar los argumentos recibidos y actualizar su estado mental.

## Evaluación de argumentos

- **Consideraciones objetivas:**
  - un argumento puede ser visto como una prueba tentativa de una conclusión, y los agentes pueden utilizar alguna tipo de convención objetiva para definir como es establecida la calidad de la prueba.
- **Consideraciones subjetivas:**
  - el agente puede considerar sus propias preferencias para evaluar un argumento.

## Generación de argumentos

- Relación entre generación de propuestas y argumentos.
- Reglas explícitas.
- Factores de influencia:
  - Autoridad
  - Utilidad esperada
  - Honestidad
  - Confianza / Credibilidad
- Depende del conocimiento del oponente.
- Inconsistencia y falta de completitud de la información.

## Generación de argumentos

- Reglas explicitas

- IF

i envió un pedido al agente  $j$  para ejecutar  $\alpha$  &

$j$  rechazó el pedido &

$j$  tiene objetivo  $g_1$  y  $g_2$  &

$j$  prefiere  $g_2$  a  $g_1$  &

haciendo  $\alpha \rightarrow \neg g_1$  &

haciendo  $\beta \rightarrow \neg g_2$  &

THEN

***threaten(i, j, not accept(j, i,  $\alpha$ , 2),  $\beta$ , 1)***

## Generación de argumentos

- Reglas explicitas

- IF

i envió un pedido al agente  $j$  para ejecutar  $\alpha$  &

$j$  rechazó el pedido &

$j$  tiene objetivo  $g_1$  y  $g_2$  &

$j$  prefiere  $g_2$  a  $g_1$  &

haciendo  $\alpha \rightarrow \neg g_1$  &

haciendo  $\beta \rightarrow g_2$  &

THEN

***reward(i, j, accept(j, i,  $\alpha$ , 2),  $\beta$ , 1)***

## Generación de argumentos

- Reglas explícitas

- IF

i envió un pedido al agente  $j$  para ejecutar  $\alpha$  &

$j$  rechazó el pedido &

$j$  prometió ejecutar  $\alpha$  en el pasado

THEN

***appeal(i, j, accept(j, i,  $\alpha$ , 2), [past\_promise(j,i,do(j,  $\alpha$ ),0)], 1)***

## Selección de argumentos

- Selección del más adecuado.
- Se asocia un peso, que indica la fuerza del argumento.
- Confiabilidad del argumento.

## Selección de argumentos

- Ordenar los argumentos por su “fuerza”:
  - 1. Apelación de practica predominante.
  - 2. Contraejemplo.
  - 3. Apelación de promesa.
  - 4. Apelación de propio interés.
  - 5. Promesa de recompensa.
  - 6. Amenaza.

## Selección de argumentos

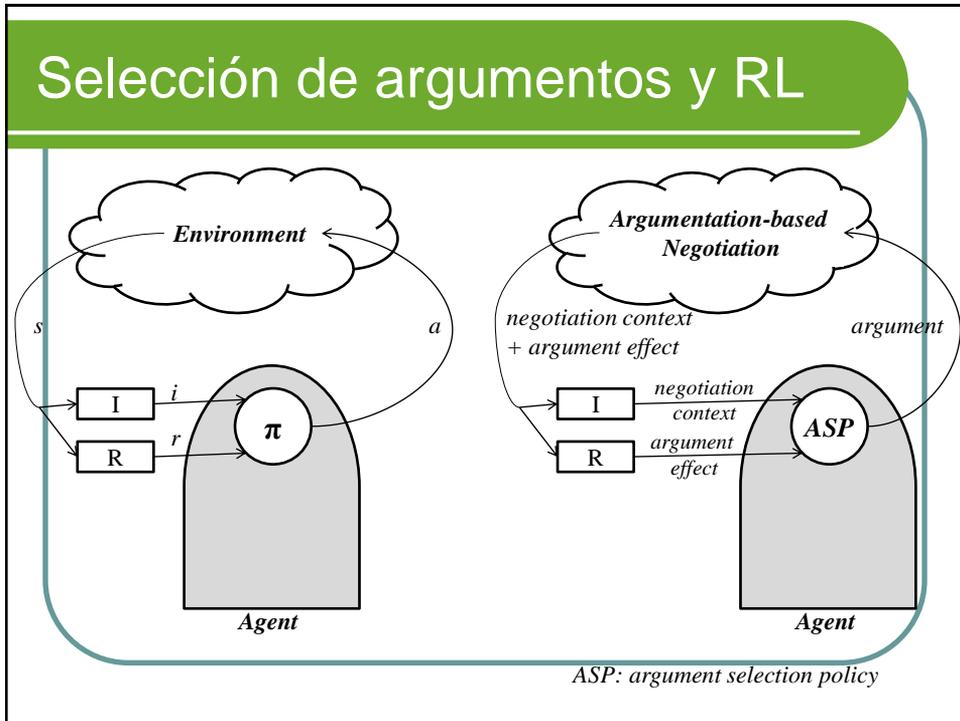
- Factor de confianza y utilidad
  - Regla 1: if **Trust** is **low** && **Utility\_proposal** is **high**  
then send a ***strong argument***
  - Regla 2: if **Trust** is **high** && **Utility\_proposal** is **low**  
then send a ***weak argument***

## Selección de argumentos

- Minimal por cardinalidad:
  - Se puede elegir el argumento más corto en tamaño porque ofrece un blanco pequeño contra el cual su oponente puede contra-argumentar.
- Minimal por conjunto de inclusión:
  - si uno de los argumentos es un super-conjunto de otro, se elige el último.
- Minimal por impacto:
  - El agente asigna factores de impacto, en un rango de 0 a 1, a cada literal que forma un argumento y así distinguir entre literales importantes de los poco importantes. El factor de impacto de un argumento es obtenido multiplicando los factores de todos sus literales.

## Aprendizaje de preferencias para la selección de argumentos

- Problemas de los enfoques tradicionales de selección de argumentos
  - No toman en cuenta
    - El proceso de aprendizaje de nuevas reglas ni la actualización de las existentes
      - Reglas generales no son buenas en contextos específicos.
      - Reglas específicas no pueden ser aplicadas a nuevos contextos.
    - La aparición de nuevos factores de que deben ser considerados por el mecanismo de selección de argumentos.



## Learning argument selection preferences

- Un algoritmo RL actualiza la política de selección de argumentos
  - $\pi$  es representado por un set de preferencias

## Preferencias

- $\text{preference}(\text{argument}(\text{TYPE}, \text{SENDER}, \text{RECEIVER}, \text{CONC}, [\text{PREM}]],[\text{CONTEXT}], S, C)$ 
  - *TYPE* es el tipo de argumento retórico
    - reward, threat, appeal;
  - *SENDER* es el agente que lo expresa;
  - *RECEIVER* es el agente a quien va dirigido;
  - *CONC* es la conclusión del argumento;
  - *PREM* es el set de premisas que soportan la conclusión;

## Preferencias

- *CONTEXT* es el set de factores del contexto en el cual el argumento fue expresado.
- Dichos factores influyen la negociación y pueden variar según el dominio.
- Cada factor es representado por una variable. Por ejemplo:
  - *utility(Ut)*: representa la utilidad asociada a una propuesta que motiva la negociación.
  - *urgency(Ur)*: representa la urgencia del emisor por alcanzar el acuerdo.
  - *trust(T)*: indica el nivel de confianza entre el emisor y el receptor.
  - *authority(A)*: indica la relación de autoridad entre el emisor y el receptor.

## Preferencias

- S es el soporte de la preferencia

$$Support(arg) = \frac{count_{arg}}{count_{tot}}$$

- Considerando  $arg$  como el argumento que representa la preferencia.
- $count_{arg}$  es el número de veces que un argumento que se correspondió con  $arg$  fue expresado por el agente.
- $count_{tot}$  es el número total de argumentos expresados por el agente.

- C es el valor de confianza

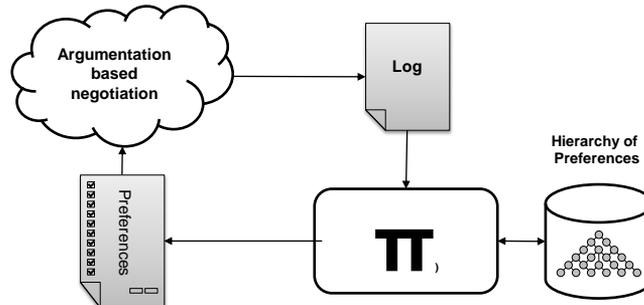
$$Confidence(arg) = \frac{success_{arg}}{count_{arg}}$$

- $success_{arg}$  es el número de veces en el cual un argumento que se corresponde con  $arg$  fue exitoso.
- Para determinar el nivel de preferencia de la regla se multiplica S por C.

## Objetivos

- Objetivo del mecanismo de aprendizaje:
  - Actualización del nivel de preferencia: actualización de los valores de soporte y confianza de las preferencias.
  - Incorporación de nuevas preferencias: se parte de un set inicial. Luego, se incorporan nuevas preferencias, más específicas, a medida que diferentes negociaciones se desarrollan.

## Proceso de aprendizaje



## Input del algoritmo

- El *input* del algoritmo de RL es el log de la negociación:
  - **a1** solicita a **a2** realizar **action1**.
  - **a2** rechaza realizar **action1**.
  - **a1** propone una recompensa diciendo “si **a2** realiza **action1**, **a1** realizará **action2**”.
  - **a2** acepta realizar **action1**.

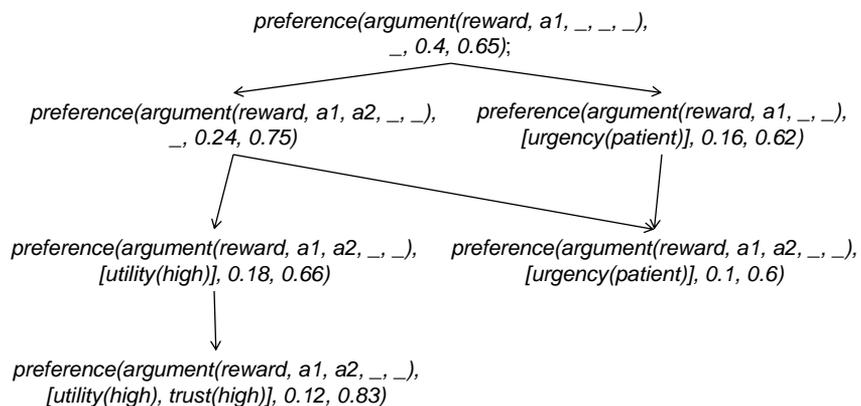
contexto: *utility(high), trust(high)*

## Input del algoritmo

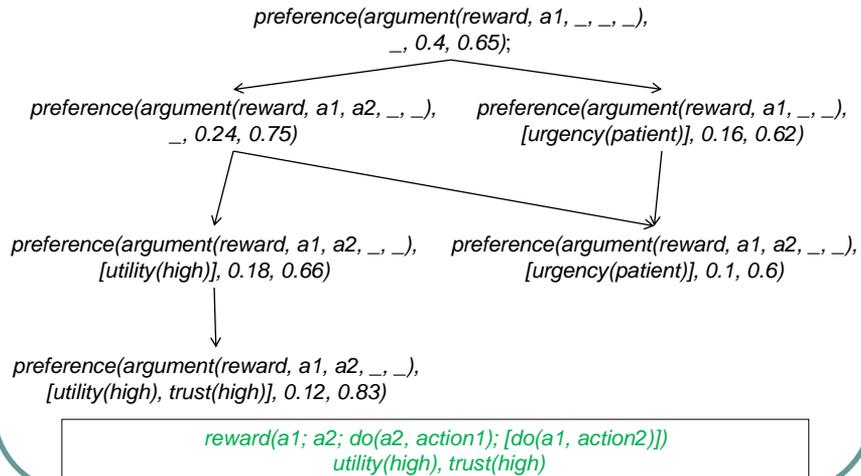
- El *input* del algoritmo de RL es el log de la negociación:
  - **a1** solicita a **a2** realizar **action1**.
  - **a2** rechaza realizar **action1**.
  - **a1** propone una recompensa diciendo “si **a2** realiza **action1**, **a1** realizará **action2**”.
  - **a2** acepta realizar **action1**. Negociación exitosa

**reward(a1; a2; do(a2, action1); [do(a1, action2)])**  
 contexto: *utility(high), trust(high)*

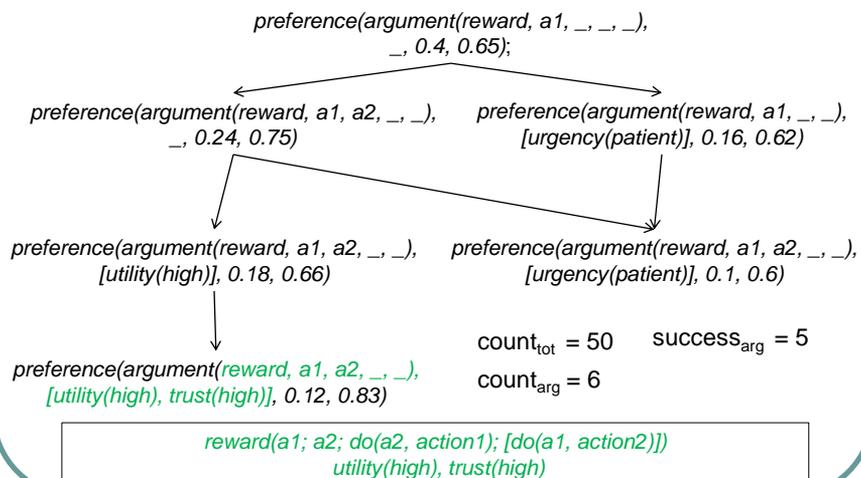
## Actualización de la política



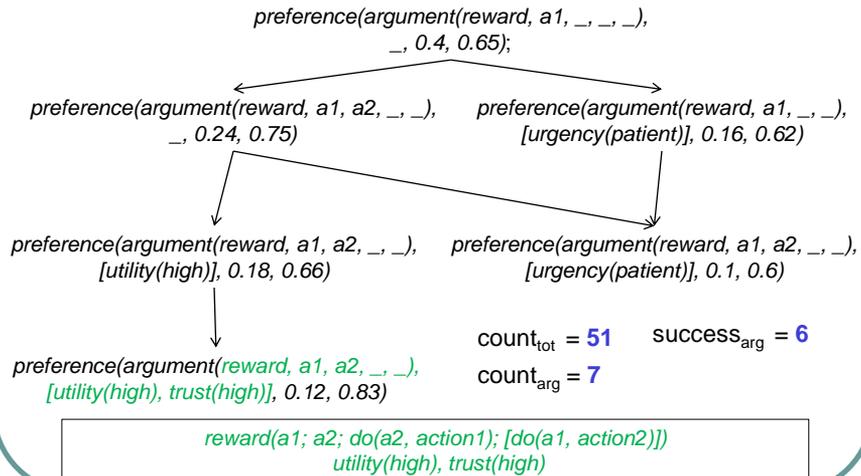
## Actualización de la política



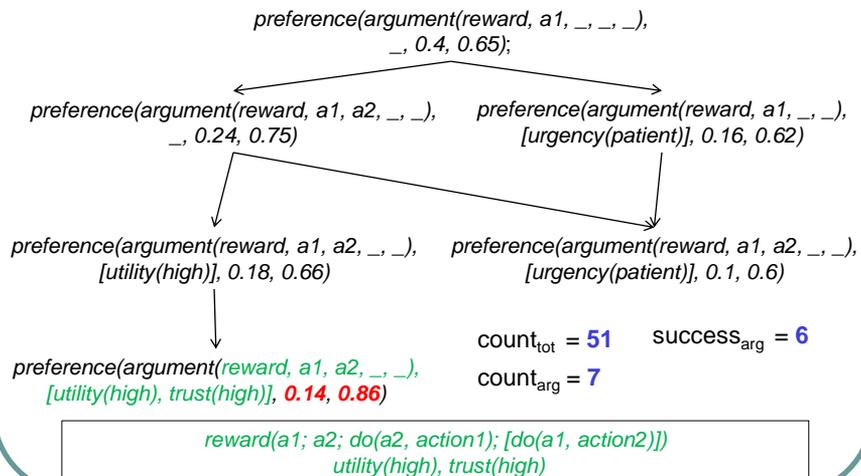
## Actualización de la política



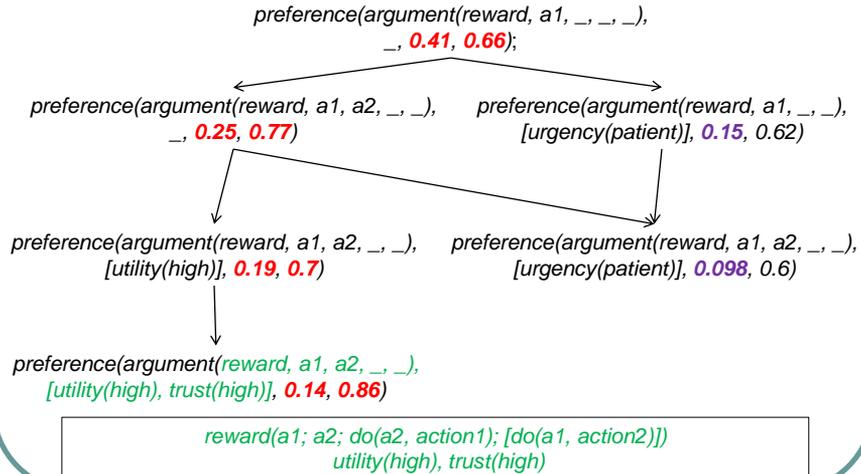
## Actualización de la política



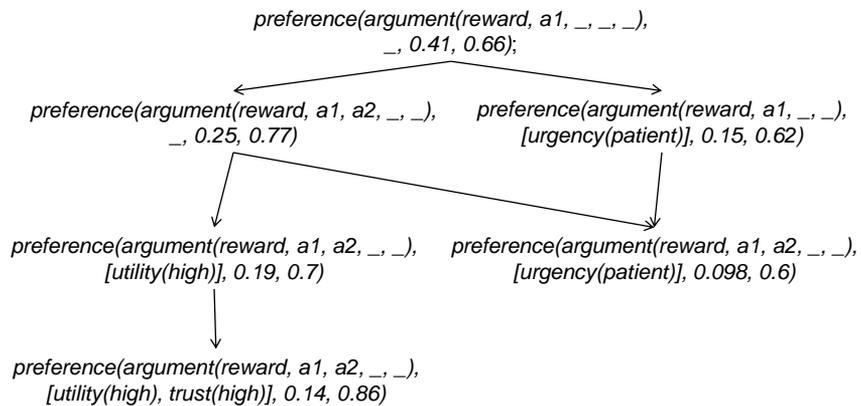
## Actualización de la política



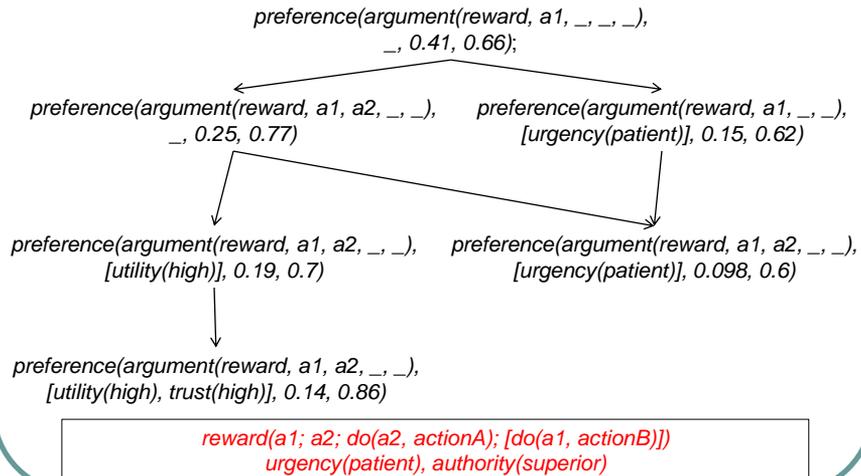
## Actualización de la política



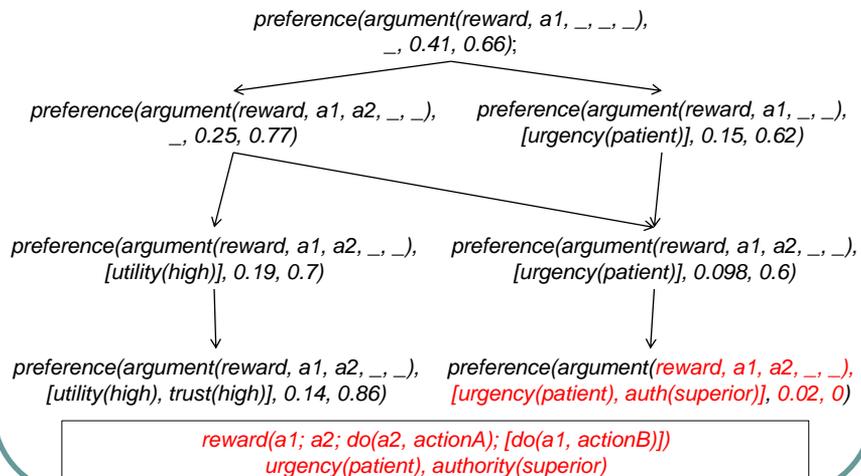
## Actualización de la política



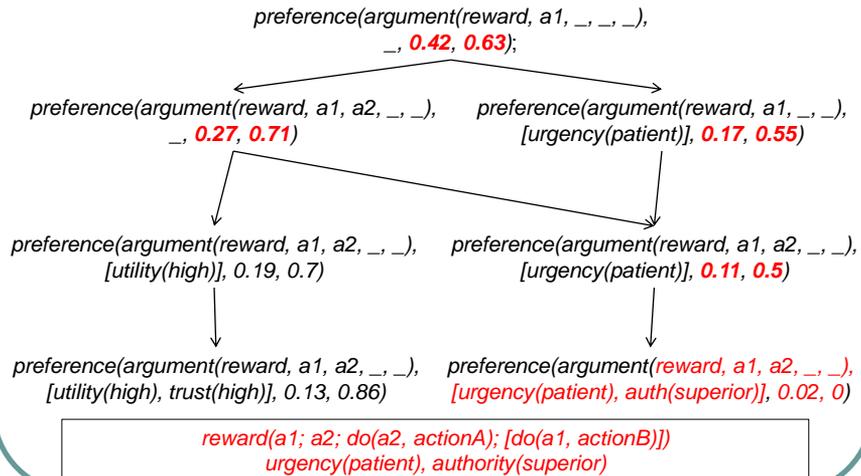
## Actualización de la política



## Actualización de la política



## Actualización de la política



## Selección de argumentos con preferencias

- Dado un set de argumentos candidatos
  - $\pi$  promedia el nivel de preferencia de cada preferencia que se corresponde con cada argumento candidato.
  - $\pi$  selecciona el argumento con media más alta.

## Anuncios Sociales

- **Optativa:**
  - Taller de Sistemas Multiagentes
    - Inicio ???
    - Cursada intensiva
    - Info: <http://www.exa.unicen.edu.ar/catedras/tmultiag/index.html>
- **Becas Doctorales CONICET**
  - Fecha presentación ???
    - Se puede deber hasta 7 finales (contando Tesis).



## Agenda

- **Agentes**
  - Definición
  - Agentes inteligentes
  - Arquitectura abstracta
    - Arquitectura BDI
    - Reinforcement learning
  - Agentes personales – móviles
- **Sistemas multiagentes**
  - Conceptos
  - Comunicación
  - Coordinación
    - Planning
    - Negociación

