Multiagent Systems

Architectures for Intelligent Agents

الدكتور/ مصطفى شبل

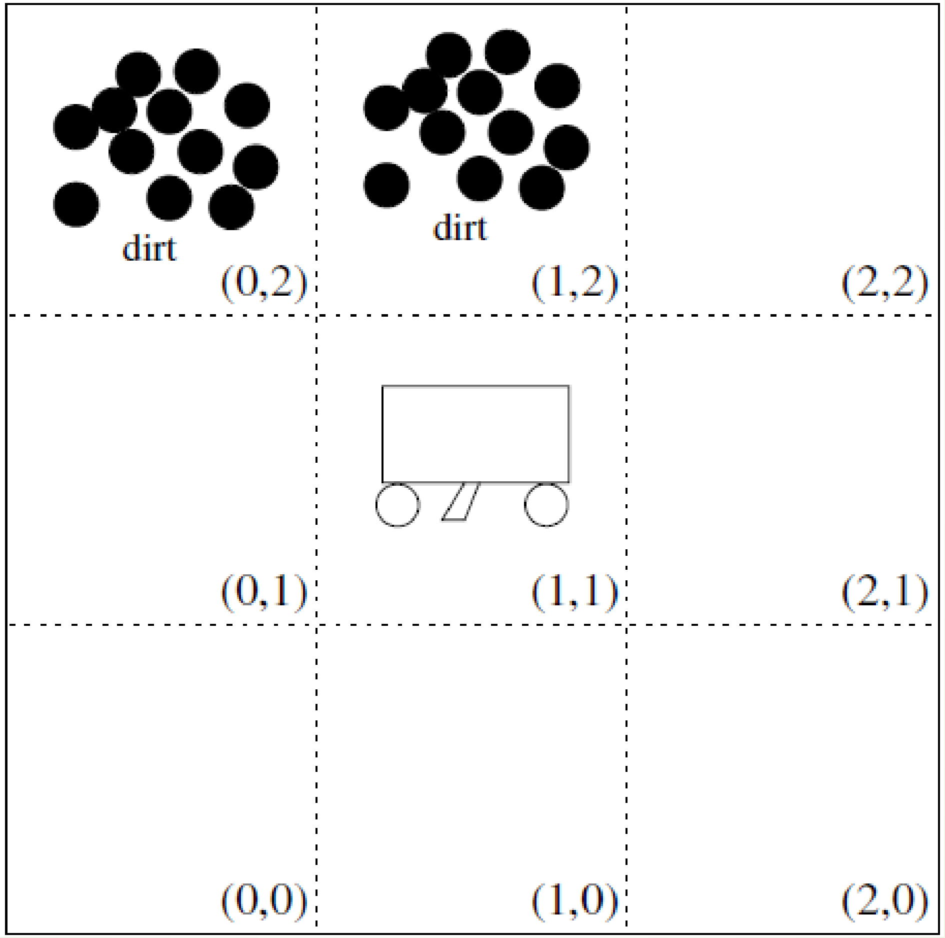
http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Architectures for Intelligent Agents

- ***LOGIC-BASED AGENTS –*** in which the decision about what action to perform is made via logical deduction;
- ***Reactive Agents*** – in which decision making is implemented in some form of direct mapping from situation to action;
- ***Belief-desire-intention Agents*** – in which decision making depends upon the manipulation of data structures representing the beliefs, desires, and intentions of the agent; and finally,
- ***Layered Architectures*** – in which decision making is realized via various software layers, each of which is more or less explicitly reasoning about the environment at different levels of abstraction.

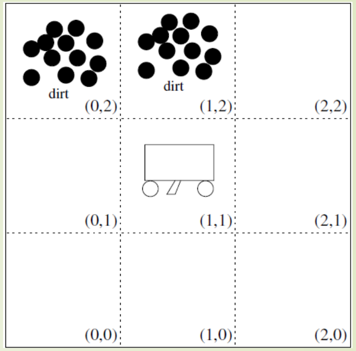
http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

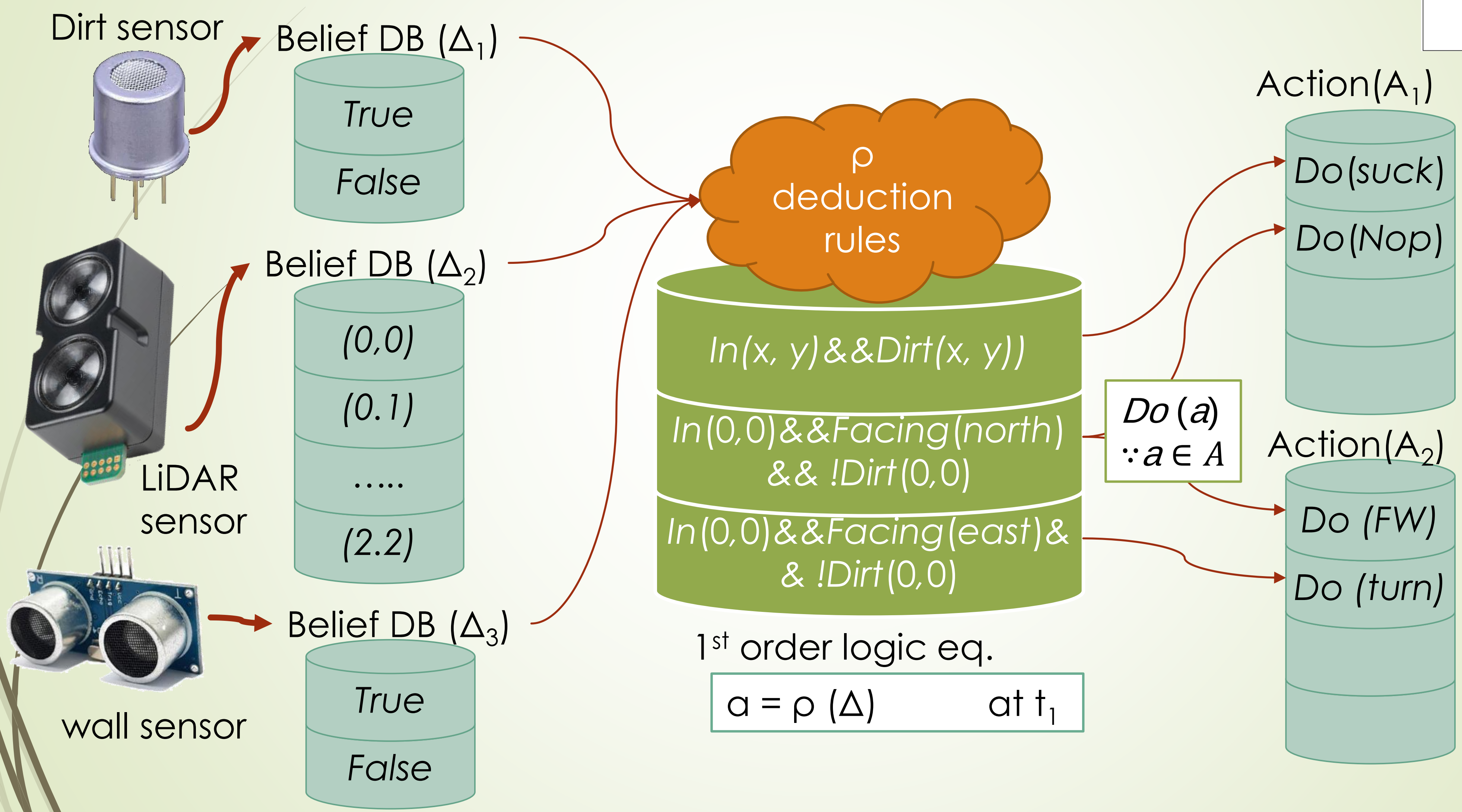
Logic-Based Architectures



http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

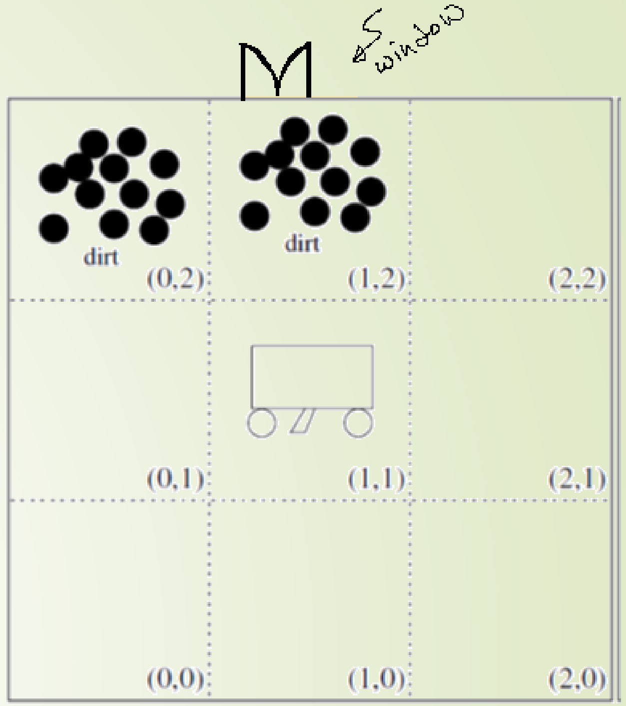
Logic-Based Architectures



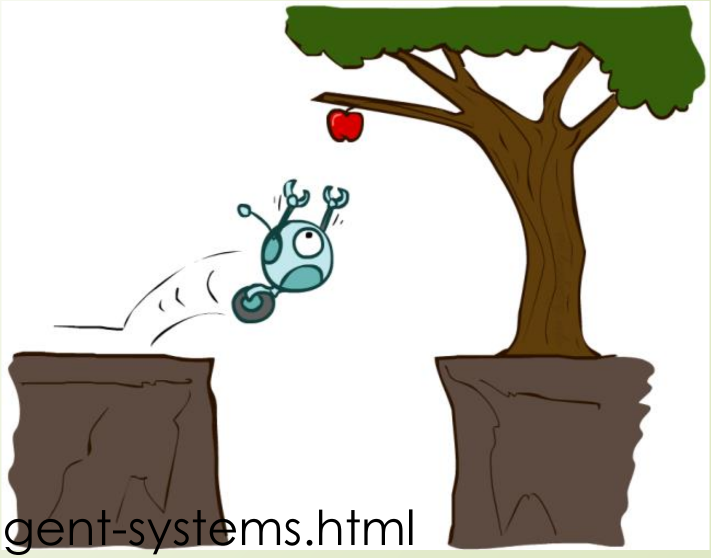


*laser imaging, detection, and ranging"

Logic-Based Architectures



- The environment has changed between t_1 and t_2 (***temporal information***), then there is no guarantee that a will still be optimal.
- An agent is said to enjoy the property of **calculative rationality (Real time processing)** if and only if its decision-making apparatus will suggest an action that was optimal when the decision-making process began.
- Problem of "translating" raw data provided by the **agent's sensors** into an internal symbolic form. (noise , threshold).
- Decision-making is viewed as deduction (inference).



http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

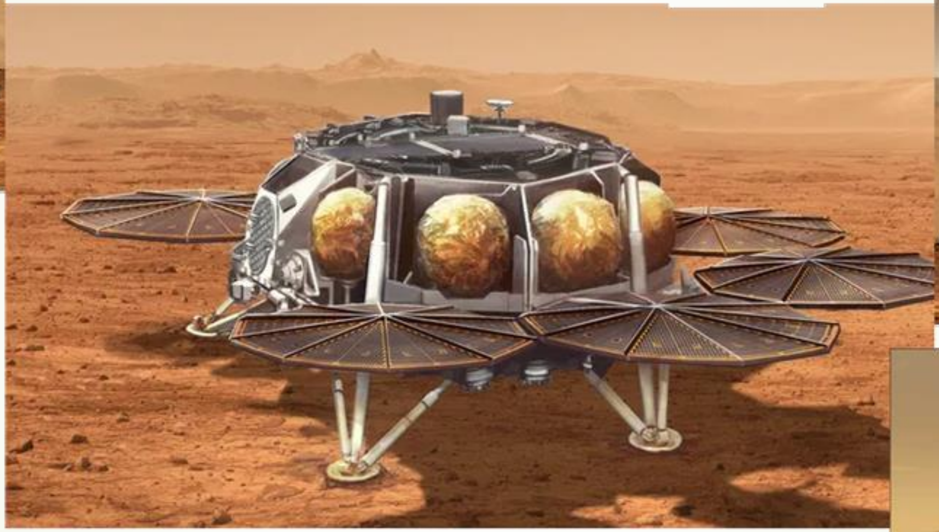
PEAS

- Consider, e.g., the task of designing an automated vacuum cleaner:
  - Performance measure: cleanness, efficiency: distance traveled to clean, battery life, security.
  - Environment: room, table, wood floor, carpet, different obstacles.
  - Actuators: wheels, different brushes, vacuum extractor.
  - Sensors: camera, dirt detection sensor, cliff sensor, bump sensors, infrared wall sensors.

http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

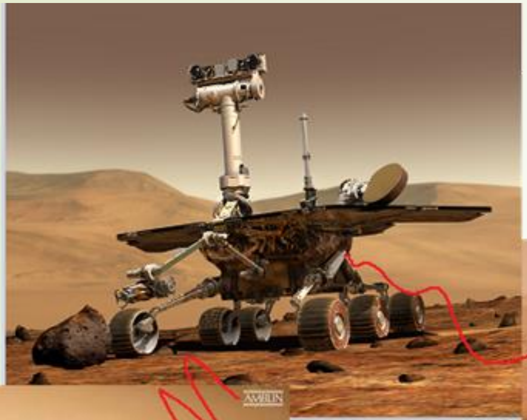
Reactive Architectures

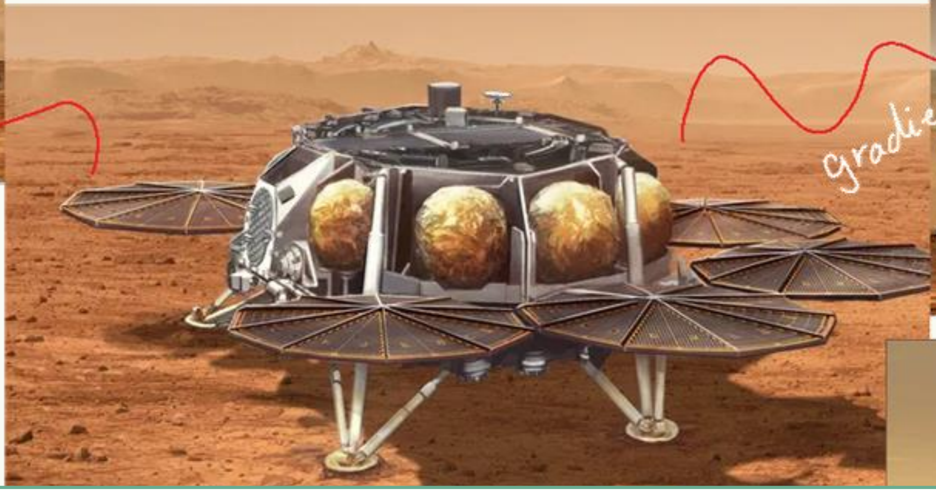
Reactive Architectures

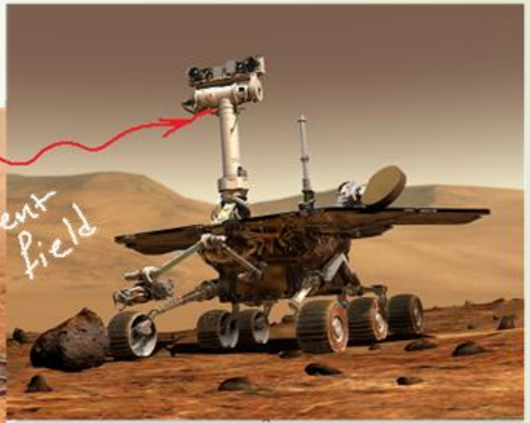


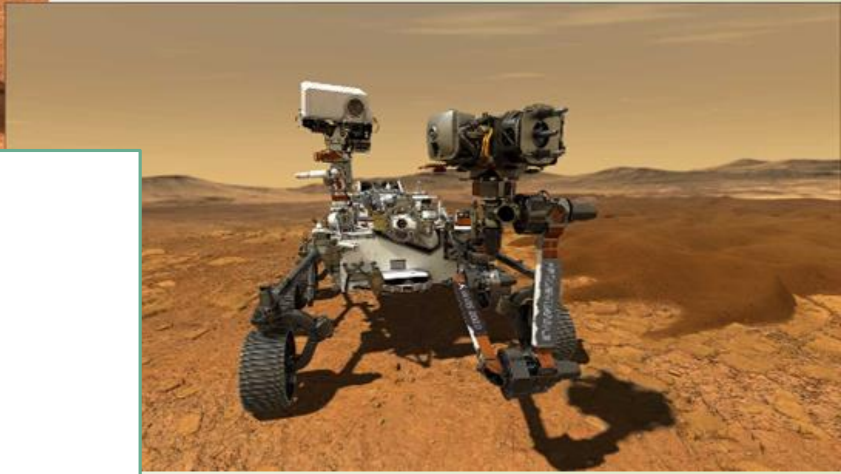
http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Reactive Architectures (The Subsumption Architecture)









- Consider, e.g., the task of designing an automated swarm:
  - Performance measure: get samples, time constrain, return to base
  - Environment: uncertain .
  - Actuators: wheels, different brushes.
  - Sensors: camera, cliff sensor, infrared wall sensors.

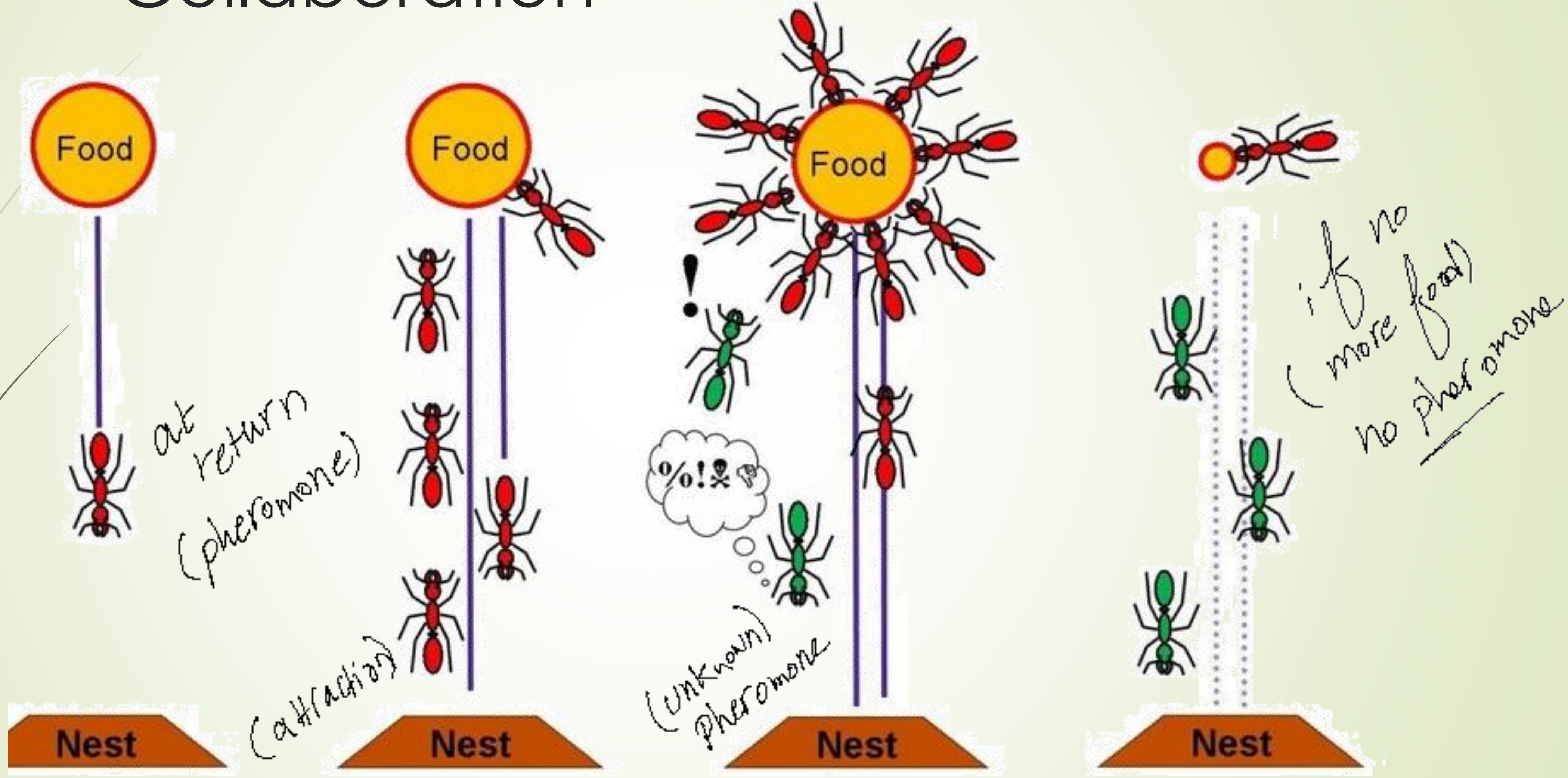
Reactive Architectures (The Subsumption Architecture)

- if true then move randomly.
- if detect a sample then pick sample up.
- if carrying samples and not at the base then travel up gradient
- if carrying samples and at the base then drop samples
- if detect an obstacle then change direction.

Priority

http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Collaboration



http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Reactive Architectures (The Subsumption Architecture)

- if true then move randomly.
- if detect a sample then pick sample up.
- if carrying samples and not at the base then travel up gradient
  - *if* carrying samples and *not* at the base *then* drop 2 crumbs *and* travel up gradient.
- if carrying samples and at the base then drop samples
- if detect an obstacle then change direction.

http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Reactive Architectures (The Subsumption Architecture)

- if true then move randomly.
- if detect a sample then pick sample up.
- *if* carrying samples and *not* at the base *then* drop 2 crumbs *and* travel up gradient.
- if carrying samples and at the base then drop samples
- if detect an obstacle then change direction.

http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Reactive Architectures (The Subsumption Architecture)

if true then move randomly.

if detect a sample then pick sample up.

if carrying samples and *not* at the base *then* drop 2 crumbs *and* travel up gradient.

if sense crumbs *then* pick up 1 crumb *and* travel down gradient.

if carrying samples and at the base then drop samples

if detect an obstacle then change direction.

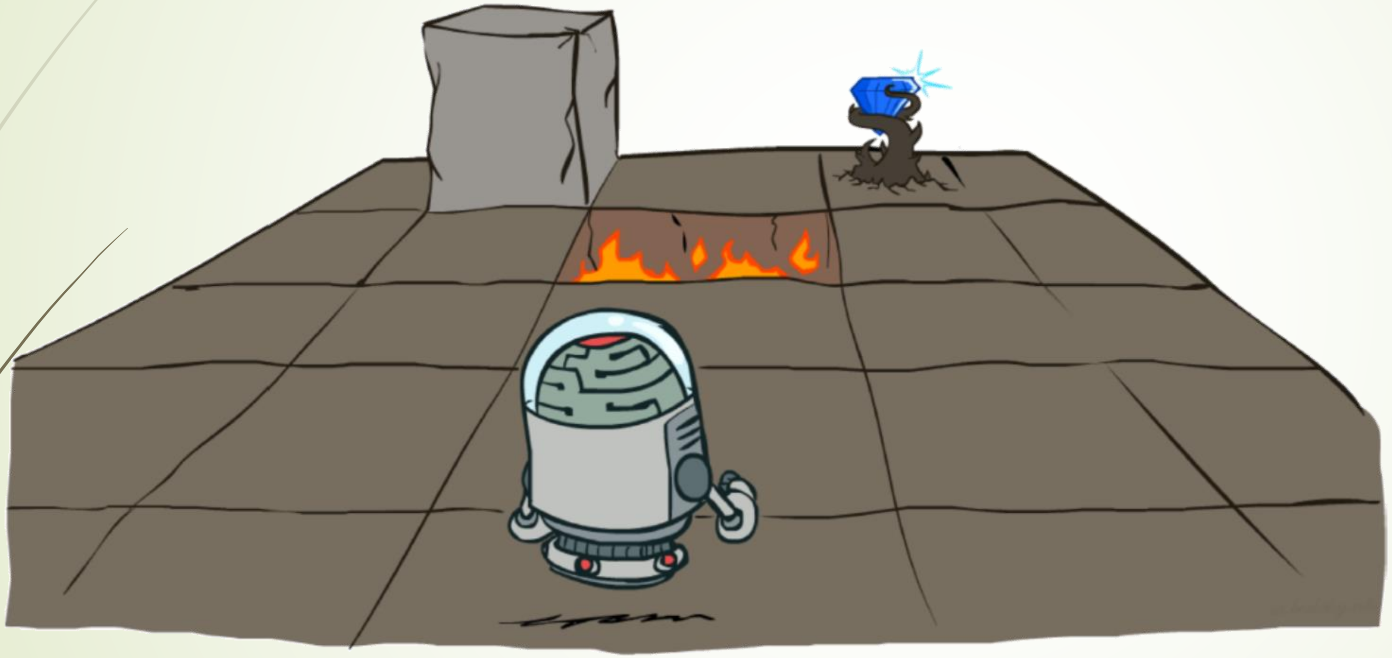
http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Disadvantages

- The dynamics of the interactions between the different behaviors become too complex to understand.
- Short term (local environment) reactive system.

http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Reactive Architectures (Markov Decision Processes)



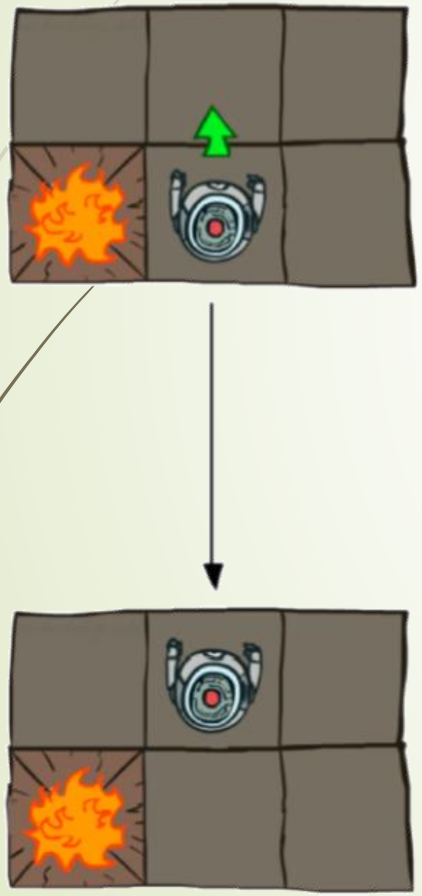
Andrey Andreyevich Markov

Born	14 June 1856 N.S. Ryazan, Russian Empire
Died	20 July 1922 (aged 66) Petrograd, Russian SFSR
Nationality	Russian
Alma mater	St. Petersburg University
Known for	Markov chains; Markov processes; stochastic processes

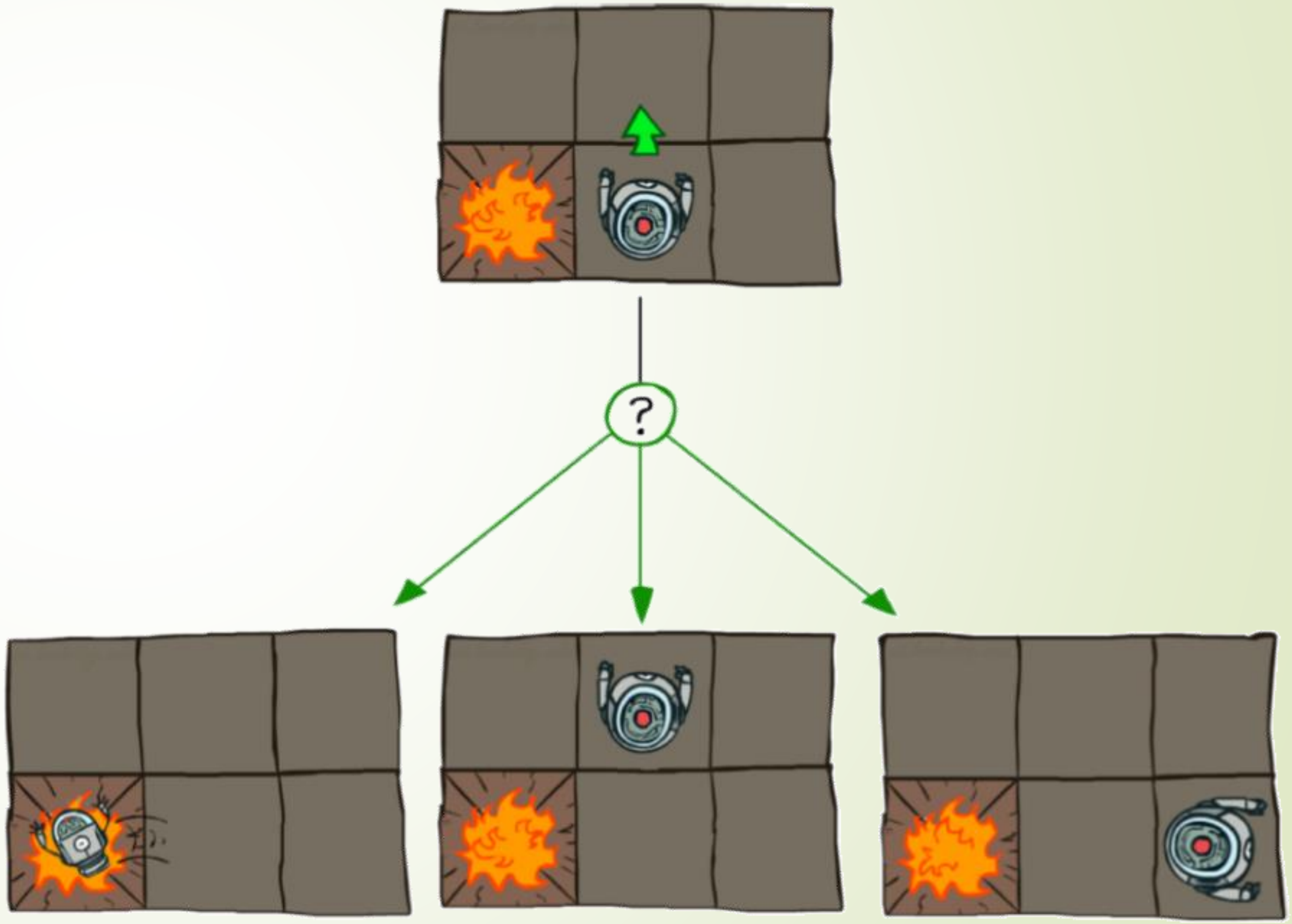
http://drshiple-courses.weebly.com/autonomous-multiagent-systems.html

Grid World Actions

Deterministic Grid World

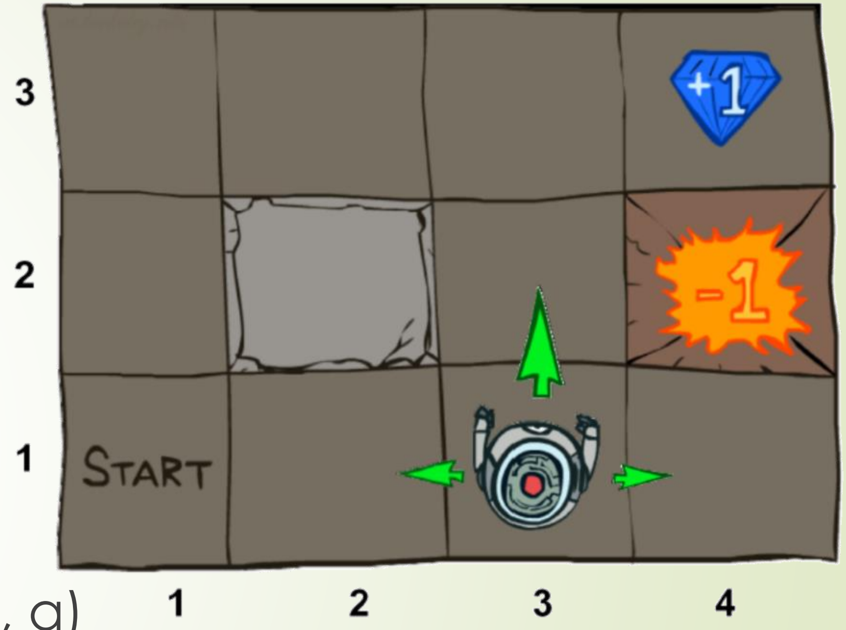


Stochastic Grid World



[These slides adapted from Anca Dragan, University of California]

Markov Decision Processes

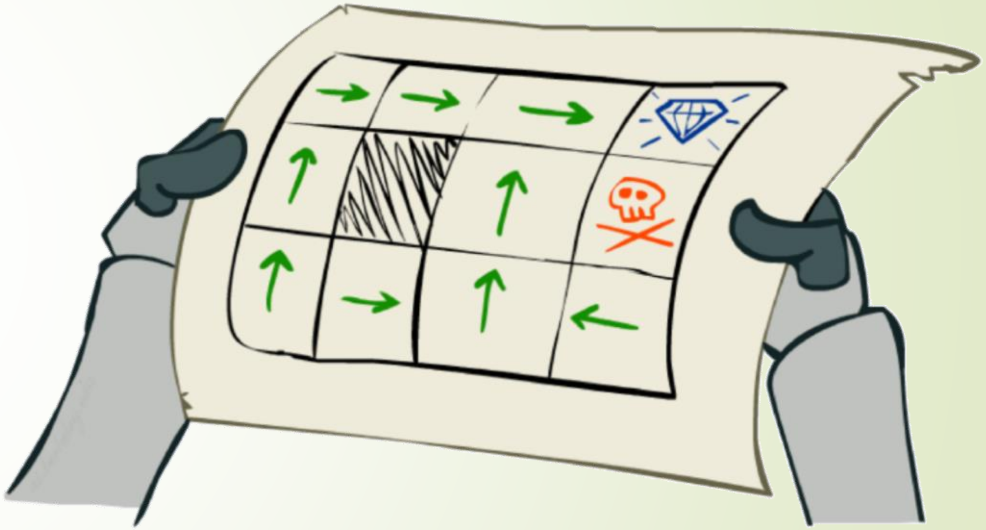


- An MDP is defined by:
- A set of states (s ∈ S);(1,2),(3,1)…etc
- A set of actions (a ∈ A); (FW, left, right, Back)
- A transition function T(s, a, s')
  - Probability that a from s leads to s', i.e., P(s' | s, a)
  - Also called the model or the dynamics
- A reward function R(s, a, s')
  - Sometimes just R(s) or R(s')
- A start state (1,1)
- Maybe a terminal state (3,3).

Policies

- For Markov decision processes, “Markov” means action outcomes depend only on the current state

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

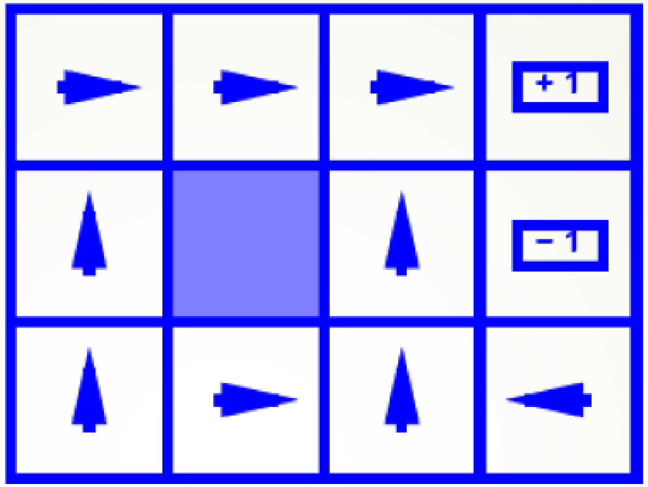


Optimal policy when R(s, a, s’) = -0.03 for all non-terminals s

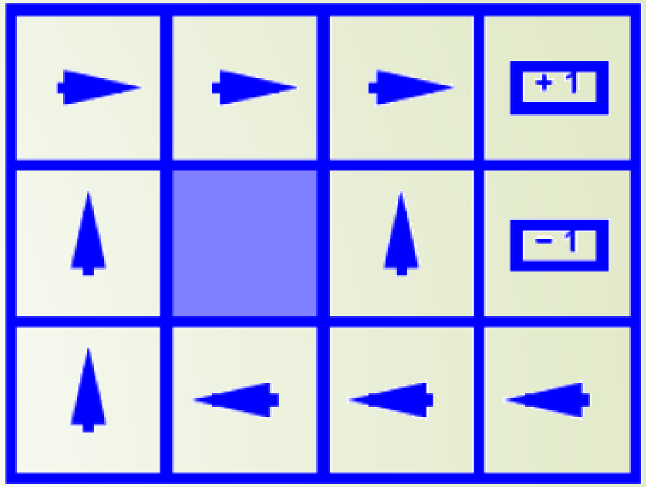
For MDPs, we want an optimal policy p*: S → A

1. A policy p gives an action for each state
2. An optimal policy is one that maximizes expected utility if followed
3. An explicit policy defines a logical based agent

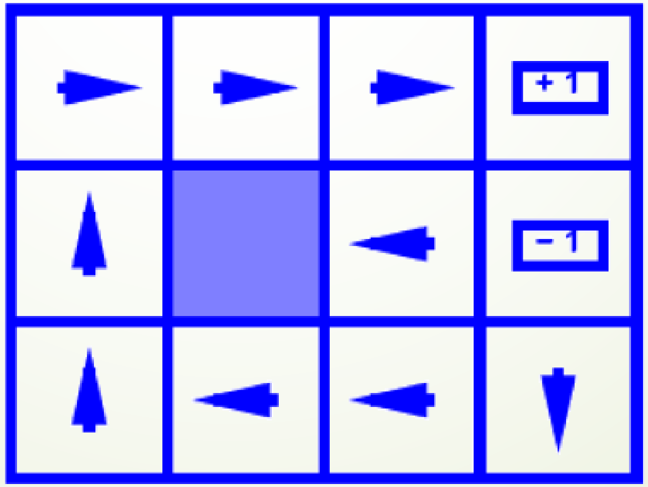
Optimal Policies



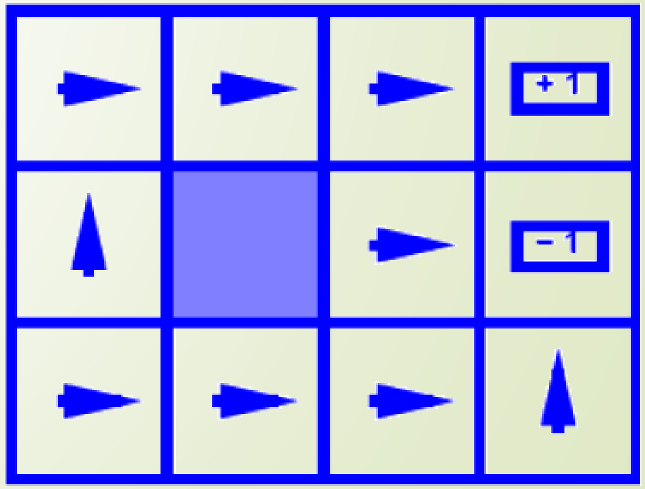
R(s) = -0.4



R(s) = -0.03

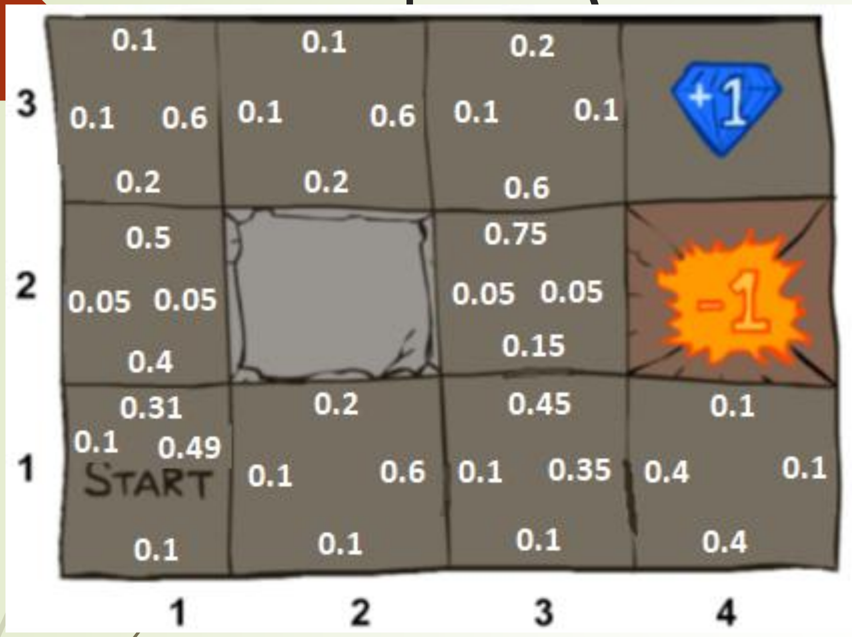


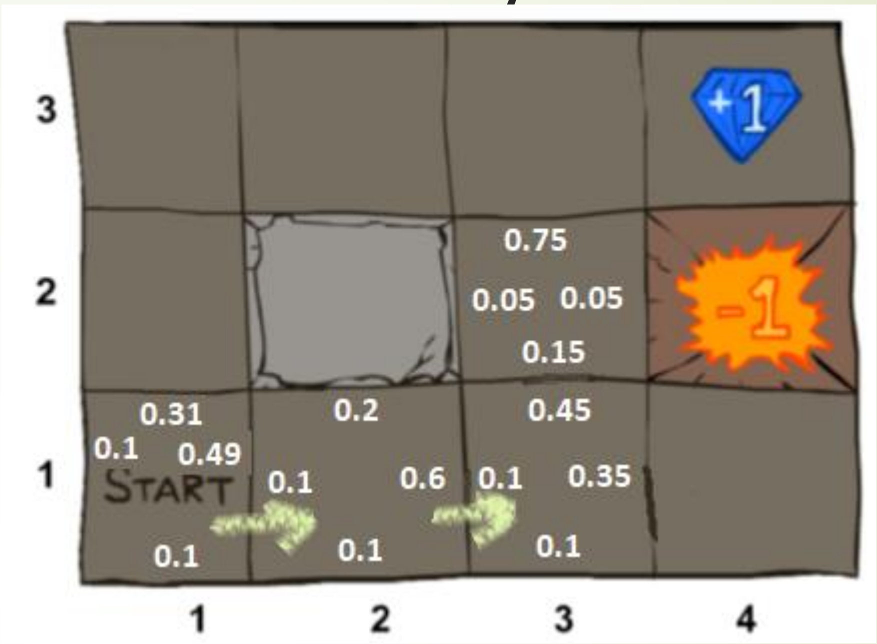
R(s) = -0.01



R(s) = -2.0

Example (deterministic env.):





- A start state (1,1)
- terminal state (1,3).

A transition function T(s, a, s')

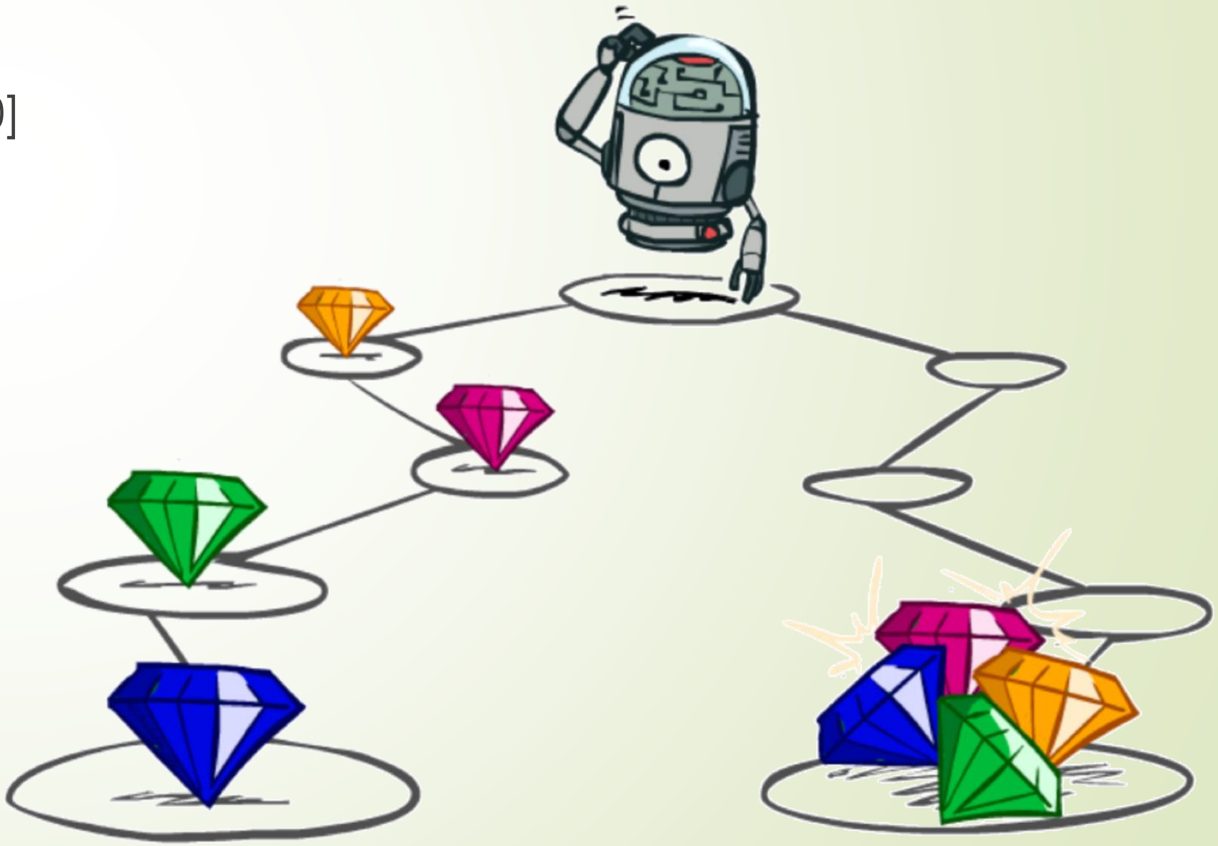
Probability that a from s leads to s', i.e., P(s' | s, a) = 0.49 *0.6 =0.29

Probability that a from s leads to s', i.e., P(s' | s, a) = 0.31*0.5*0.6*0.6*0.6*0.15 =0.005

Utilities of Sequences

- What preferences should an agent have over reward sequences?
- More or less? [1, 2, 2] or [1, 1, 3]

- Now or later? [0, 0, 1] or [1, 0, 0]



Discounting

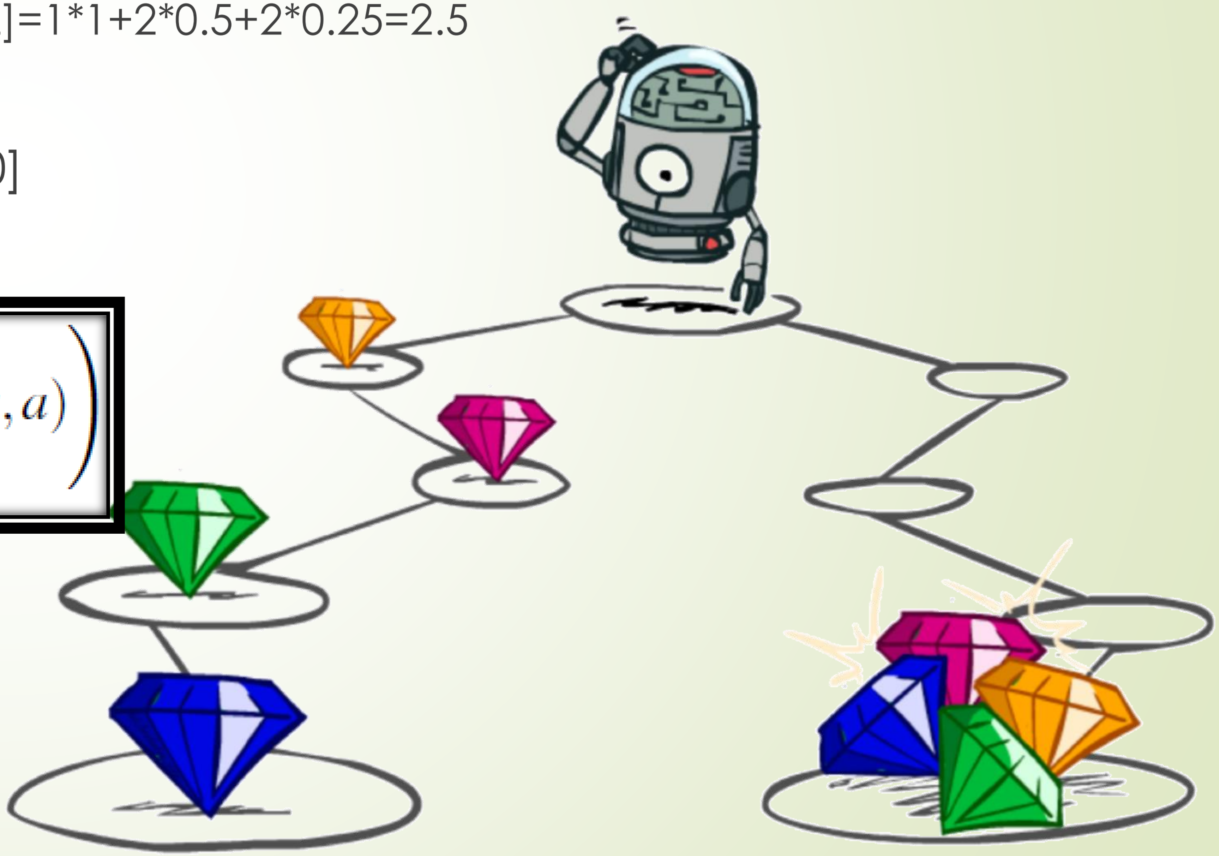
- It's reasonable to maximize the sum of rewards
- It's also reasonable to prefer rewards now to rewards later
- One solution: values of rewards decay exponentially

- Policy = Choice of action for each state
- Utility = sum of (discounted) rewards

Utilities of Sequences

- What preferences should an agent have over reward sequences?
- More or less? [1, 2, 2] or [1, 1, 3]
  - sol:- for discount = 0.5 → [1, 2, 2]=1*1+2*0.5+2*0.25=2.5
  - [1, 1, 3] =1*1+1*0.5+3*0.25=2.25
- Now or later? [0, 0, 1] or [1, 0, 0]

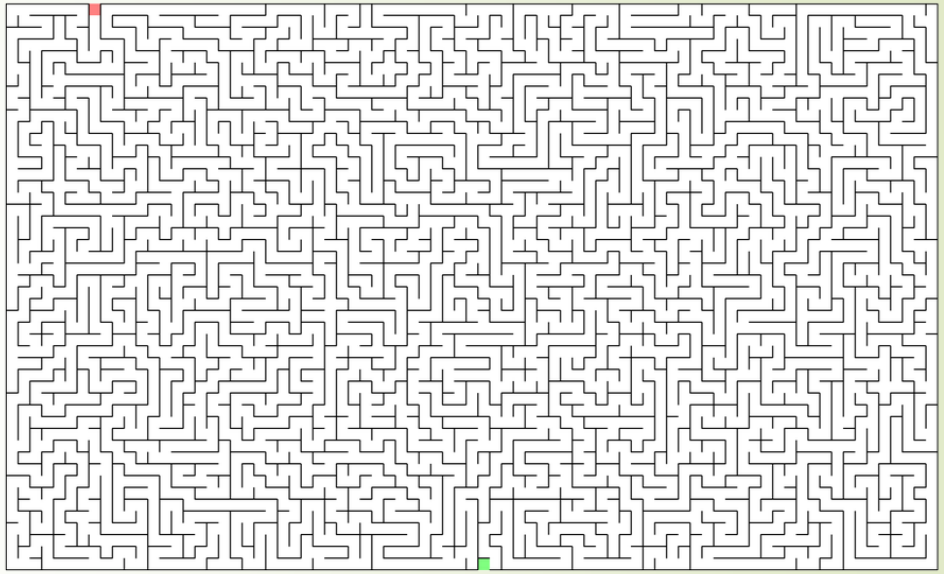
$$d^*(s) = \arg\max_{a \in A} \left(r(s,a) + \lambda \sum_{s' \in S} p(s' \mid s,a) \right)$$



Belief-Desire-Intention Architectures

Un-deterministic env.

Intention : to reach the final destination
Persist : keep trying to achieve my intentions
Believe : after how many trails ur believe in success still standing

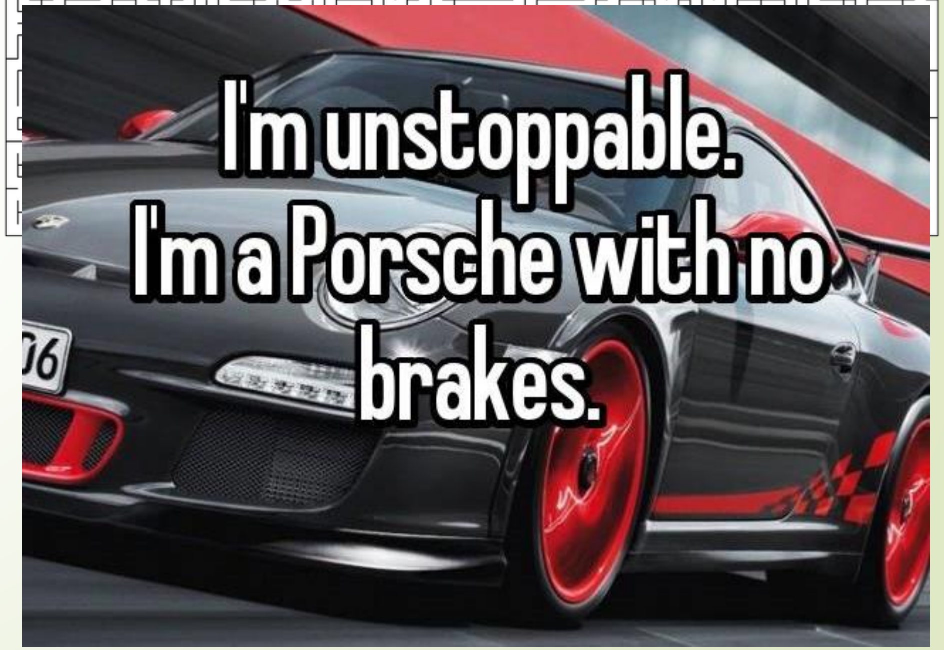


bold agents
(those that never stop to reconsider

cautious agents
(those that are constantly stopping to reconsider)

Un-deterministic env.

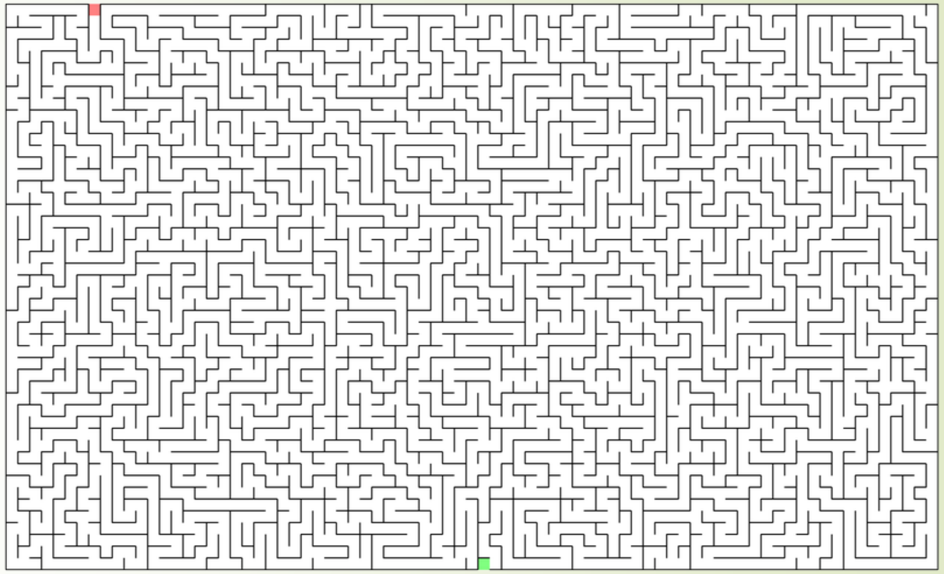
Intention : to reach the final destination
Persist : keep trying to achieve my intentions
Believe : after how many trails ur believe in success still standing



bold agents
(those that never stop to reconsider

Un-deterministic env.

Intention : to reach the final destination
Persist : keep trying to achieve my intentions
Believe : after how many trails ur believe in success still standing



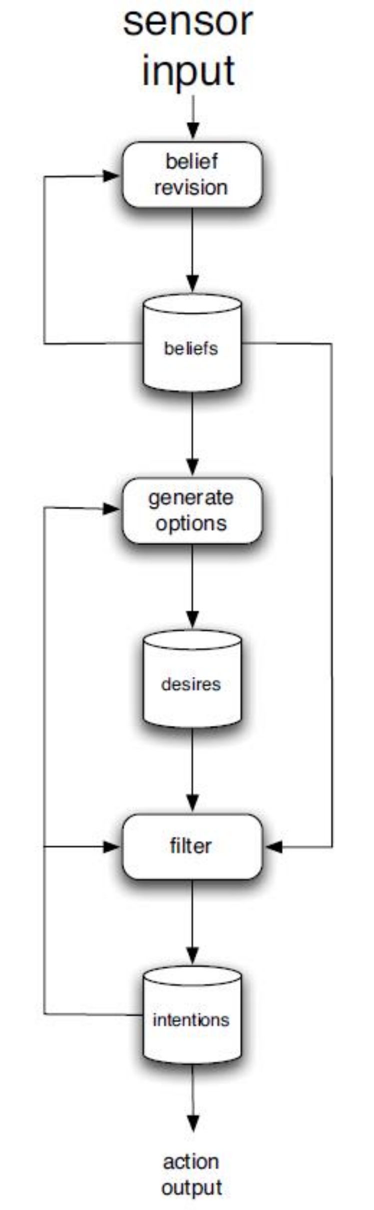
bold agents
(those that never stop to reconsider

cautious agents
(those that are constantly stopping to reconsider)

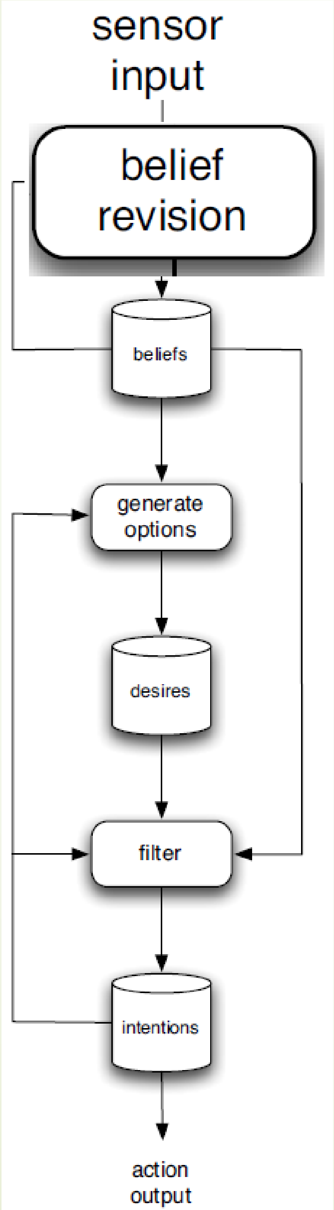
rate of world change, γ $\begin{cases} \gamma\ low = \text{the environment does not change quickly (} \\ \gamma\ high = \text{the environment changes frequently} \end{cases}.$

γ ↓:purely proactive, goal directed behavior is adequate
dynamic environments, the ability to react to changes by modifying intentions becomes more important

Schematic diagram of a generic belief-desire-intention architecture

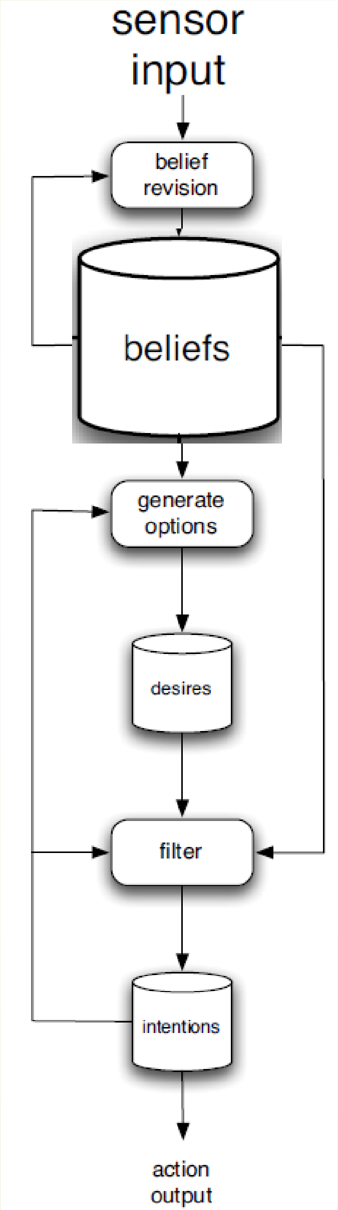


Schematic diagram of a generic belief-desire-intention architecture



which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;

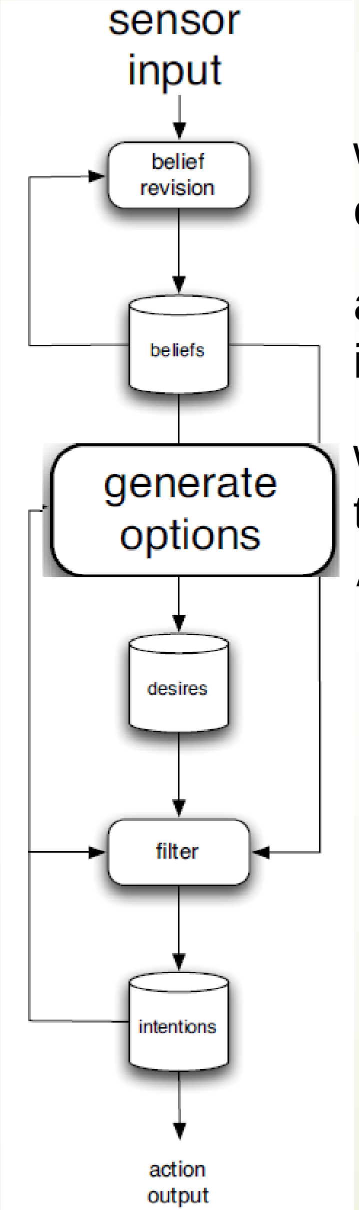
Schematic diagram of a generic belief-desire-intention architecture



which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;

a set of current *beliefs*, representing information the agent has about its current environment;

Schematic diagram of a generic belief-desire-intention architecture

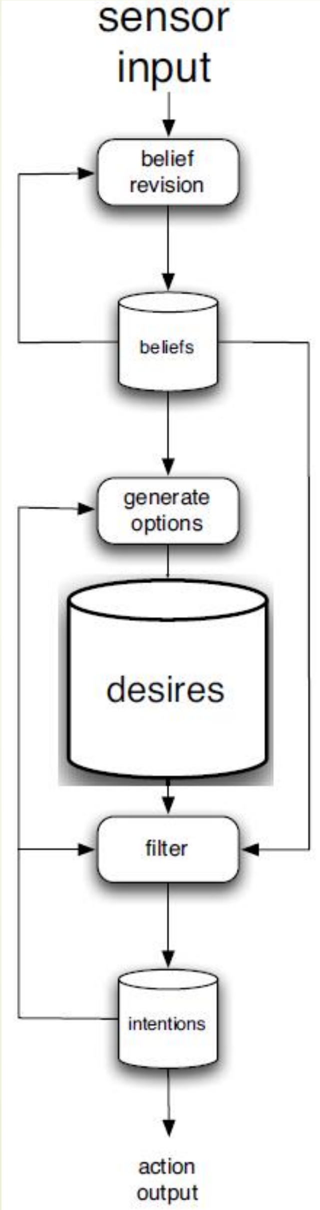


which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;

a set of current *beliefs*, representing information the agent has about its current environment;

which determines the options available to the agent (its desires), on the basis of its current beliefs about its environment and its current *intentions*;

Schematic diagram of a generic belief-desire-intention architecture



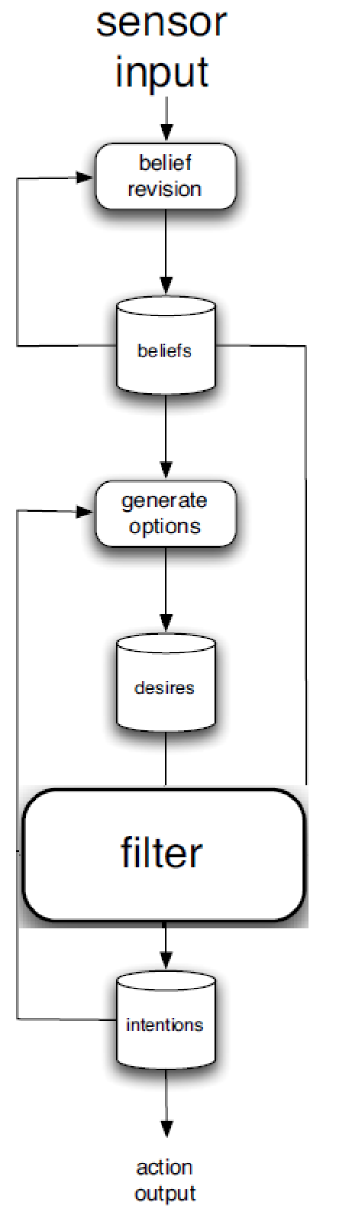
which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;

a set of current *beliefs*, representing information the agent has about its current environment;

which determines the options available to the agent (its desires), on the basis of its current beliefs about its environment and its current *intentions*;

representing possible courses of actions available to the agent;

Schematic diagram of a generic belief-desire-intention architecture



which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;

a set of current *beliefs*, representing information the agent has about its current environment;

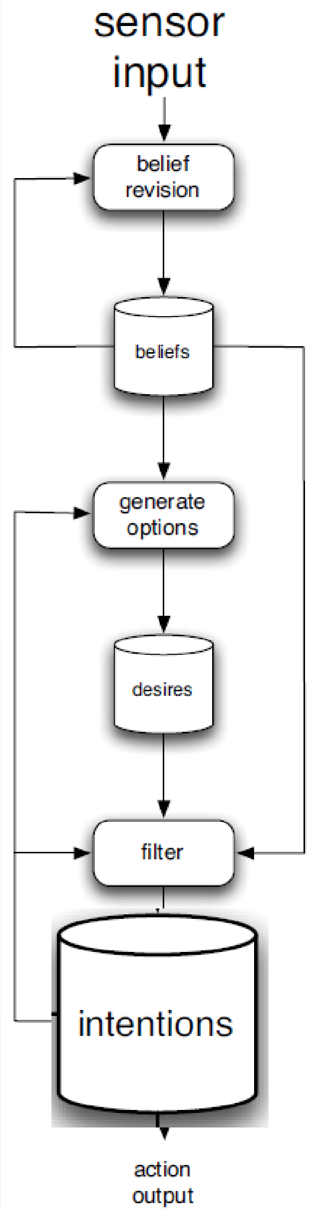
which determines the options available to the agent (its desires), on the basis of its current beliefs about its environment and its current *intentions*;

representing possible courses of actions available to the agent;

which represents the agent's *deliberation* process, and which determines the agent's intentions on the basis of its current beliefs, desires, and intentions;

- *Drop*
  - intentions that are no longer achievable
  - expected cost exceeds the expected gain
- *retain*
  - not achieved *and* has positive overall benefit
  - *adopt* new intentionsexploit new opportunities

Schematic diagram of a generic belief-desire-intention architecture



which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;

a set of current *beliefs*, representing information the agent has about its current environment;

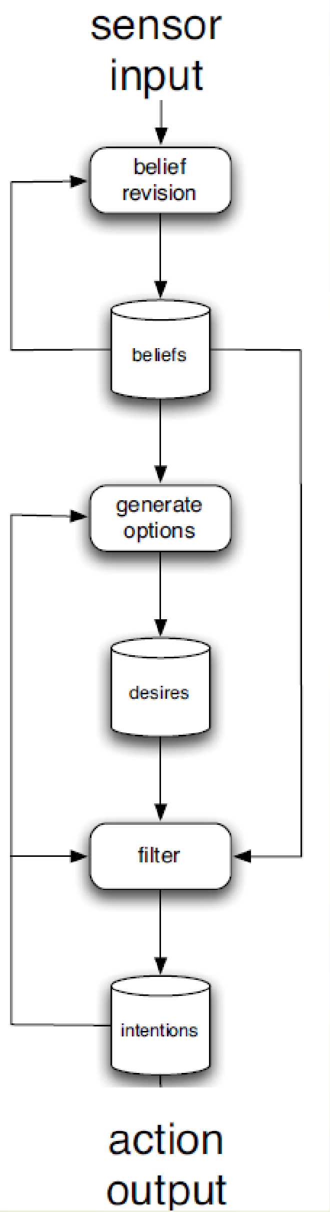
which determines the options available to the agent (its desires), on the basis of its current beliefs about its environment and its current *intentions*;

representing possible courses of actions available to the agent;

which represents the agent's *deliberation* process, and which determines the agent's intentions on the basis of its current beliefs, desires, and intentions;

representing the agent's current focus – those states of affairs that it has committed to trying to bring about;

Schematic diagram of a generic belief-desire-intention architecture



which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;

a set of current *beliefs*, representing information the agent has about its current environment;

which determines the options available to the agent (its desires), on the basis of its current beliefs about its environment and its current *intentions*;

representing possible courses of actions available to the agent;

which represents the agent's *deliberation* process, and which determines the agent's intentions on the basis of its current beliefs, desires, and intentions;

representing the agent's current focus – those states of affairs that it has committed to trying to bring about;

which determines an action to perform on the basis of current intentions.

Multiagent System (MAS)

- **Definition :** A multiagent system is one that consists of a number of agents, which *interact* with one another, typically by exchanging messages through some computer network infrastructure.
- Developer task : these agents will thus require the ability to ***cooperate***, ***coordinate***, and ***negotiate.***

Agent Design

- How do we build agents that are capable of independent, autonomous action in order to successfully carry out the tasks that we delegate to them?

Society Design

How do we build agents that are capable of interacting (cooperating, coordinating, negotiating) with other agents in order to successfully carry out the tasks that we delegate to them, particularly when the other agents cannot be assumed to share the same interests/goals?

Society Design

Readings

- Chapter 1:- Gerhard Weiss, Multiagent Systems, second edition