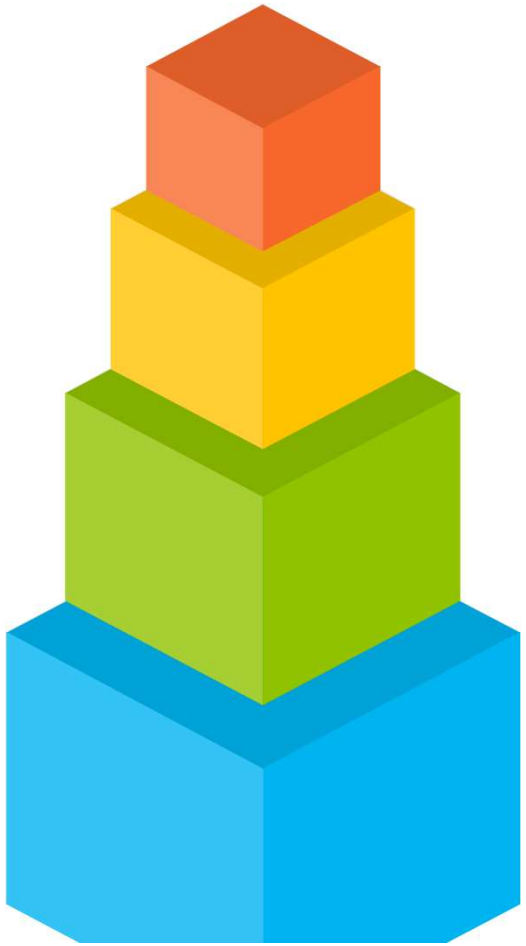
Intelligent Agents: Abstract Architectures

Prof. Iván Axel Dounce Nava



Overview

3. Abstract Architectures
4. Performance metrics
5. Agent synthesis

3. Abstract Architectures

The mathematical fundamentals of agents

Next…

- Formal definition of agent and environment.
- Tropistic Agents
- Perception in Agents
- Internal state in Agents

Formal definition of "Agent" and "Environment"

- Lets first answer the following:
  - How to represent environment states?
  - How to represent actions?
  - What are "Runs"?
  - What is a state transformer function? History dependent agents!
- And then:
  - What is an environment?
  - What is an agent?
- Finally…
  - What is a system? All coming together.
  - Formal run

Formal definition of “Agent” and “Environment”

Purely reactive agents

- Instead of a "**state transform**" function, we define a simple "**action**" function:
  - That generates an action from a single state.
  - This means the agent is not history dependent.

Adding Perception to agents

- How can an agent "see" its environment?
- Is it not enough to input data?
  - Short answer… NO!

- Important rule for agents: you are not allowed to mess with their minds!

Adding Perception to agents

- Let's define a function called "see".
- And a function called "action".

Adding Internal states

- Let's add some memory to our agent.
- And now we can redefine our previous functions…

4. Performance Metrics

And how do we know that the agent is doing good?

Agents are involved in a delegated task

- How can we measure performance concerning the task?
  - 1. Utility over state
  - 2. Utility over runs
  - 3. Predicate Task Environments

Utility over state

- Associating a value to a state.
- But… What happens if we need to see long term utility?

Utility over runs

- Associates a value to a run
- Inherently long term

Example of utility

- Tileworld

Expected Utility

- Good practice to validate if your agents will have good utility, beforehand.

- We want to find an Optimal Agent:
  - The agent with the best (on average) expected utility

Predicate Task Specification

- What if the utility is only 1 or 0, Win or Loose, Succeed or Fail?

- Let's define a Task Environment:
  - It tells us the properties of an environment
  - And the criteria for the agent to succeed in this environment
- Then… when did the agent succeed?
- And what would it be the probability of success?

5. Agent Synthesis

Can an environment have a successful agent?

Automatic programming

- How can we be sure a task environment will have a successful agent?

- That's why we need a synthesis algorithm:
  - That is sound
  - That is complete

Thank you for your attention