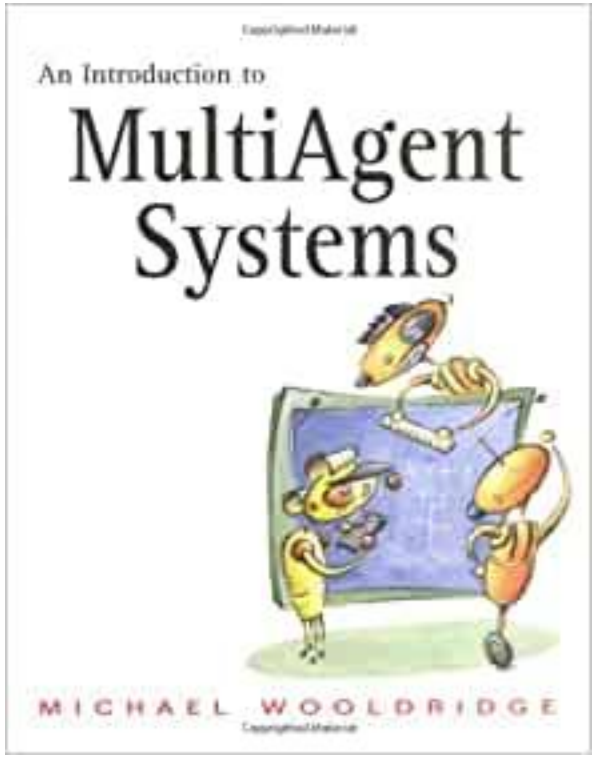
Deductive agents

Madalina Croitoru

University of Montpellier



Based on "An Introduction to MultiAgent Systems" by Michael Wooldridge, John Wiley & Sons, 2002.
http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

Agent Architectures

- An *agent* is a computer system capable of *flexible autonomous action*…
- Issues one needs to address in order to build agent-based systems…
- Three types of agent *architecture*:
  - symbolic/logical
  - reactive
  - hybrid

Agent Architectures

- Originally (1956-1985), pretty much all agents designed within AI were *symbolic reasoning* agents
- Problems with symbolic reasoning led to a reaction against this — the so-called *reactive agents* movement, 1985–present
- From 1990-present, a number of alternatives proposed: *hybrid* architectures, which attempt to combine the best of reasoning and reactive architectures

Symbolic Reasoning Agents

- The classical approach to building agents is to view them as a particular type of knowledge-based system
- This paradigm is known as *symbolic AI*
- We define a <u>deliberative</u> agent or agent architecture to be one that:
  - contains an explicitly represented, symbolic model of the world
  - makes decisions (for example about what actions to perform) via symbolic reasoning

Symbolic Reasoning Agents

- Two key (yet unsolved) problems to be solved:

1. *The transduction problem*:
that of translating the real world into an accurate, adequate symbolic description, in time for that description to be useful…vision, speech understanding, learning

2. *The representation/reasoning problem*:
how to symbolically represent information about complex real-world entities and processes, and how to get agents to reason with this information in time for the results to be useful…knowledge representation, automated reasoning, automatic planning

Deductive Reasoning Agents

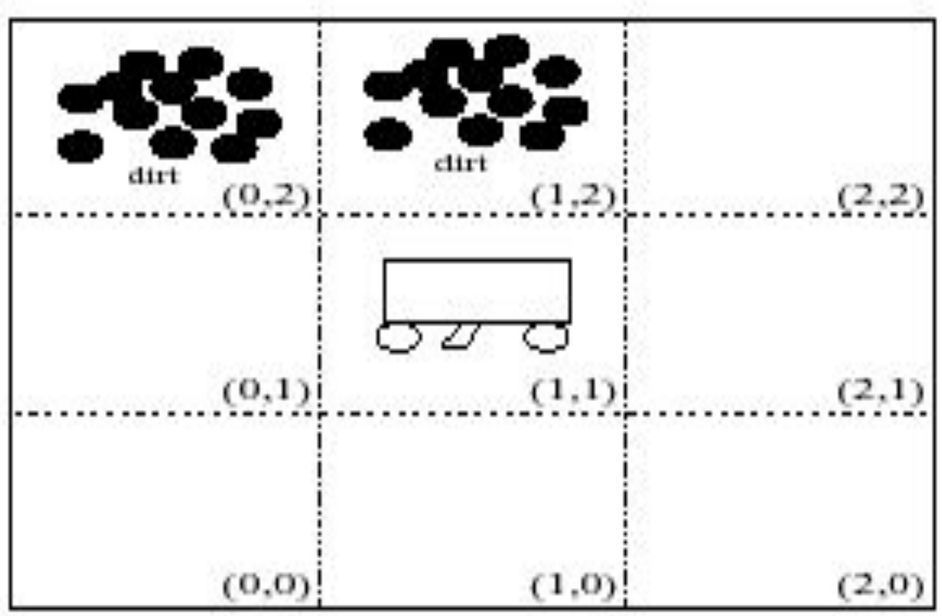
- How can an agent decide what to do using theorem proving?
- Basic idea is to use logic to encode a theory stating the *best* action to perform in any given situation
- Let:
  - ρ be this theory (typically a set of rules)
  - Δ be a logical database that describes the current state of the world
  - Ac be the set of actions the agent can perform
  - $\Delta \vdash_{\rho} \varphi$ mean that φ can be proved from Δ using ρ

Deductive Reasoning Agents

```
/* try to find an action explicitly prescribed */
for each a ∈ Ac do
        if Δ |ρ Do(a) then
            return a
        end-if
end-for
/* try to find an action not excluded */
for each a ∈ Ac do
        if Δ |ρ ¬Do(a) then
            return a
        end-if
end-for
return null /* no action found */
```

Deductive Reasoning Agents

- An example: The Vacuum World
- Goal is for the robot to clear up all dirt



Deductive Reasoning Agents

- Use 3 *domain predicates* to solve problem:

$In(x, y)$	agent is at (x, y)
$Dirt(x, y)$	there is dirt at (x, y)
$Facing(d)$	the agent is facing direction d

- Possible actions:

$$Ac = \{turn, forward, suck\}$$

P.S. *turn* means "turn right"

Deductive Reasoning Agents

- Rules ρ for determining what to do:

$$\begin{aligned}
In(0,0) \wedge Facing(north) \wedge \neg Dirt(0,0) &\longrightarrow Do(forward) \\
In(0,1) \wedge Facing(north) \wedge \neg Dirt(0,1) &\longrightarrow Do(forward) \\
In(0,2) \wedge Facing(north) \wedge \neg Dirt(0,2) &\longrightarrow Do(turn) \\
In(0,2) \wedge Facing(east) &\longrightarrow Do(forward)
\end{aligned}$$

- …and so on!
- Using these rules (+ other obvious ones), starting at $(0, 0)$ the robot will clear up dirt

Deductive Reasoning Agents

- Problems:
  - How to convert video camera input to $Dirt(0, 1)$?
  - Decision making assumes a *static* environment: *calculative* rationality
- Even where we use *propositional* logic, decision making in the worst case means solving co-NP-complete problems (PS: co-NP-complete = bad news!)
- Typical solutions:
  - weaken the logic
  - use symbolic, non-logical representations
  - shift the emphasis of reasoning from *run time* to *design time*

Exercise for this week

- Choose one scenario from last week homework
- Try to implement it using a symbolic approach using the imperative programming language of your choice
  - What problems do you see combining imperative programming and deductive agents?
- (Optional) Try and implement the same scenario using Prolog. Is this solution easier than the imperative programming one?