Deductive Reasoning Agents

Based on "An Introduction to MultiAgent Systems" and slides by Michael Wooldridge

Agent Architectures

- An agent architecture is a software design for an agent
- Top-level decomposition into perception – state – decision – action
- An agent architecture defines:
    - key data structures
    - operations on data structures
    - control flow between operations

Types of Agents

- 1950's – now: *Symbolic Reasoning Agents*
  Its purest expression, proposes that agents use explicit logical reasoning in order to decide what to do
- 1980's – now: *Reactive Agents*
  Problems with symbolic reasoning led to a reaction against this — led to the reactive agents movement
- 1990's – now: *Hybrid Agents*
  Hybrid architectures attempt to combine the best of symbolic and reactive architectures

Deductive Reasoning Agents

- Traditional approach to build AI systems (*symbolic AI*)
  - Symbolic representation of environment and behavior
  - Syntactic manipulation of symbolic representation
- Symbolic representation $\rightarrow$ *logical formulae*
- Syntactic manipulation $\rightarrow$ *logical deduction* (*theorem proving*)

Transduction Problem

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

Representation/Reasoning Problem

- The problem of 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

Agents as Theorem Provers

- Theory of agency φ – some theory that explains how an intelligent agnet should behave to optimize some performance measure
- Theory φ is viewed as executable specification that is directly executed in order to produce the agent's behavior

Agents as Theorem Provers

Deliberate agents – simple model of logic-based agents

- Internal state assumed to be a database of formulae (predicate logic)

Example

Open(valve221)
Temperature(reactor4726, 321)
Pressure(tank776, 28)

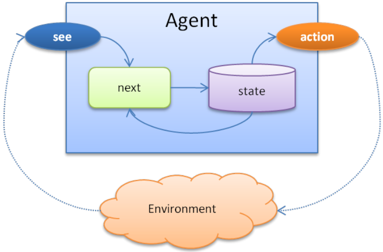
- Analogous to *beliefs* in humans – internal state may include incorrect (outdated, invalid) information

Agents as Theorem Provers

Let:

- L be the set of formulae of classical first-order logic
- $D = \mathbf{2}^L$ be the set of L databases
  - DB, DB_1, ... are the members of D
  - DB represents the internal state of an agent
- ρ is a set of deduction rules that models the agent's decision making process
- $DB \vdash_\rho \varphi$ means that the formula φ can be proved from the database DB using only the deduction rules ρ

Agents with State



- Perception function *see*
- Next state function *next*
- Action-selection function *action*

Agents as Theorem Provers

- Perception function

$$see : E \to Per$$

- Next function

$$next : D \times Per \longrightarrow D$$

- Action-selection function (defined in terms of the agent's deduction rules)

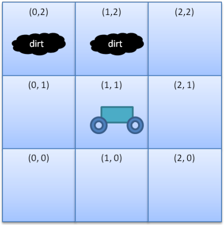
$$action : D \to Ac$$

Action Selection as Theorem Proving

```
foreach α ∈ Ac do
    if DB ⊢ρ Do(α) then
        return α;
    end
end
foreach α ∈ Ac do
    if DB ⊬ρ ¬Do(α) then
        return α;
    end
end
```

Example – Vacuum World

- A small robotic agent that cleans up a room divided into a grid of equally sized squares
- Agent is equipped with a dirt sensor and a vacuum cleaner
- Agent always has a definite orientation (north, south, east, west)
- Agent can move foreward one step and turn right 90º
- For simplicity we assume the room is 3×3 and agent always starts in square 0,0 facing north



Representing Vacuum World

- Possible percepts
  $Per = \{dirt, null\}$
- Domain predicates describing internal state
  $In(x, y)$ – agent is at (x, y)
  $Dirt(x, y)$ – there is dirt at (x, y)
  $Facing(d)$ – agent is facing direction d
- Possible actions
  $Act = \{forward, suck, turn\}$

Reasoning in Vacuum World

Deduction rules

1. $In(x, y) \wedge Dirt(x, y) \longrightarrow Do(suck)$
2. $In(0, 0) \wedge Facing(north) \wedge \neg Dirt(0, 0) \longrightarrow Do(forward)$
3. $In(0, 1) \wedge Facing(north) \wedge \neg Dirt(0, 1) \longrightarrow Do(forward)$
4. $In(0, 2) \wedge Facing(north) \wedge \neg Dirt(0, 2) \longrightarrow Do(turn)$
5. $In(0, 2) \wedge Facing(east) \wedge \neg Dirt(0, 2) \longrightarrow Do(forward)$
6. $\cdots$

Problems with Deductive Reasoning

- How to convert video camera input to $\{dirt, null\}$ and how to represent properties of dynamic environment
- Decision making assumes a static environment: *calculative rationality*
  - Decision making process suggests an action that was optimal when the process started
  - If decision making is immediate, then we can dicard this problem
- Decision making via theorem proving is complex (it may never terminate)

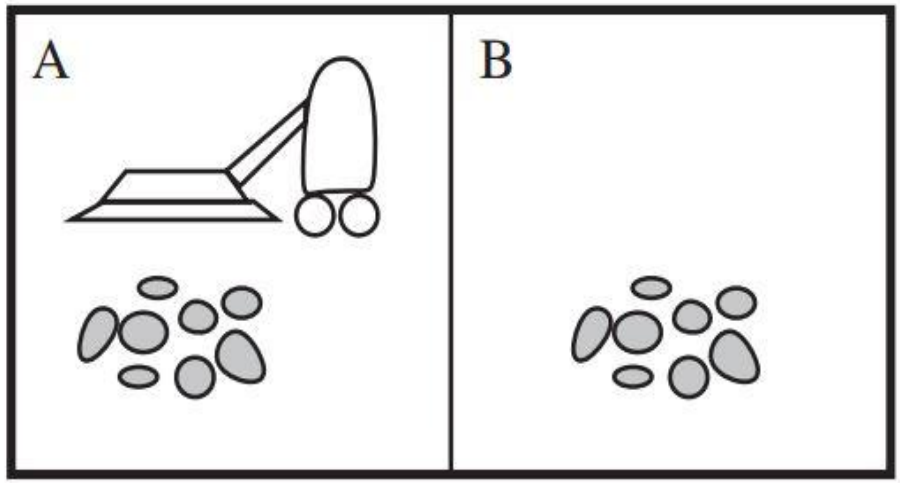
EXAMPLE: VACUUM WORLD

Formulating a problem

The real world is absurdly complex.

We solve problems by defining appropriate abstractions.

Example: Vacuum world

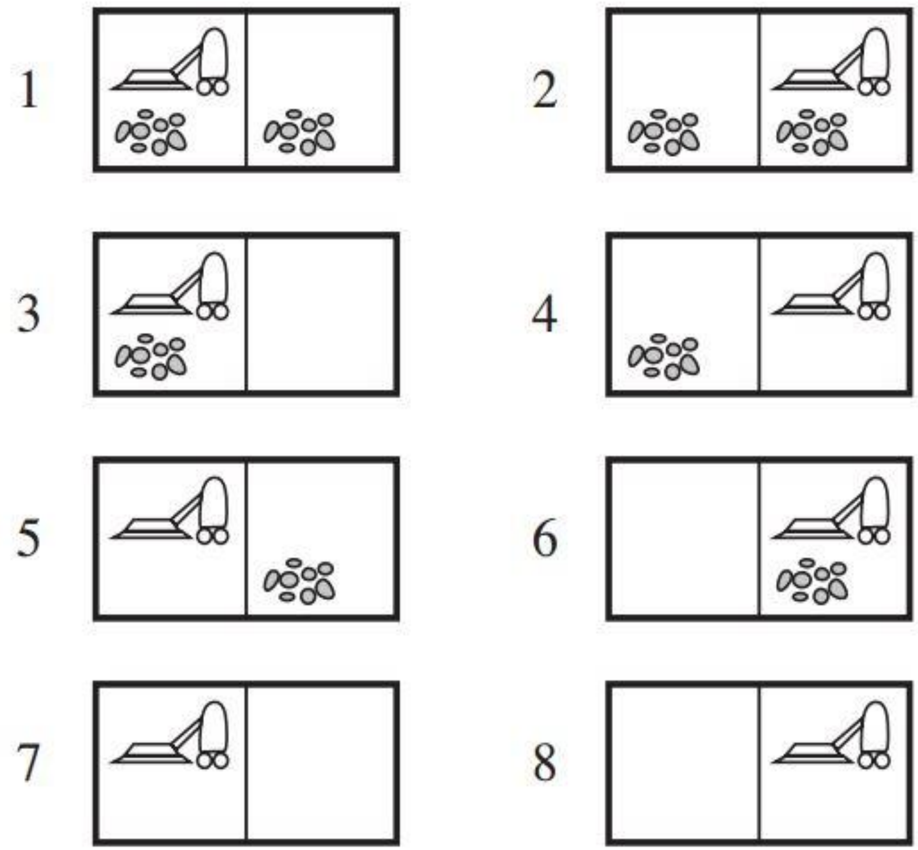


Percepts: Location and room contents, e.g., [A, Dirty]

Actions: Left, Right, Suck, NoOp

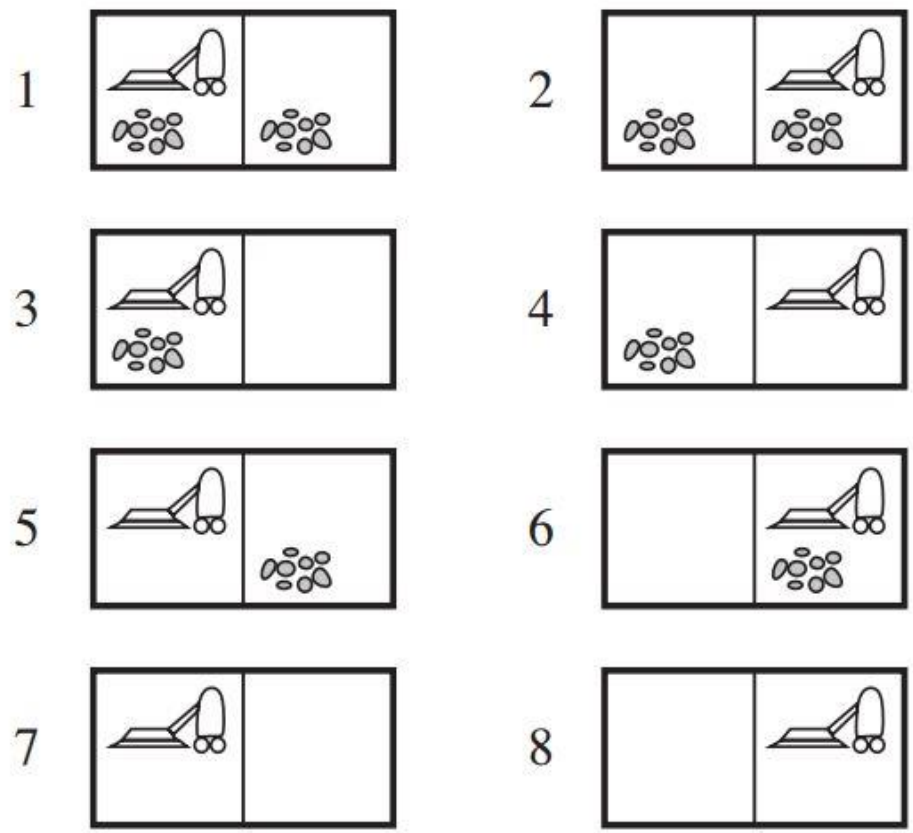
Example: Vacuum world

Complete state space



Example: Vacuum world

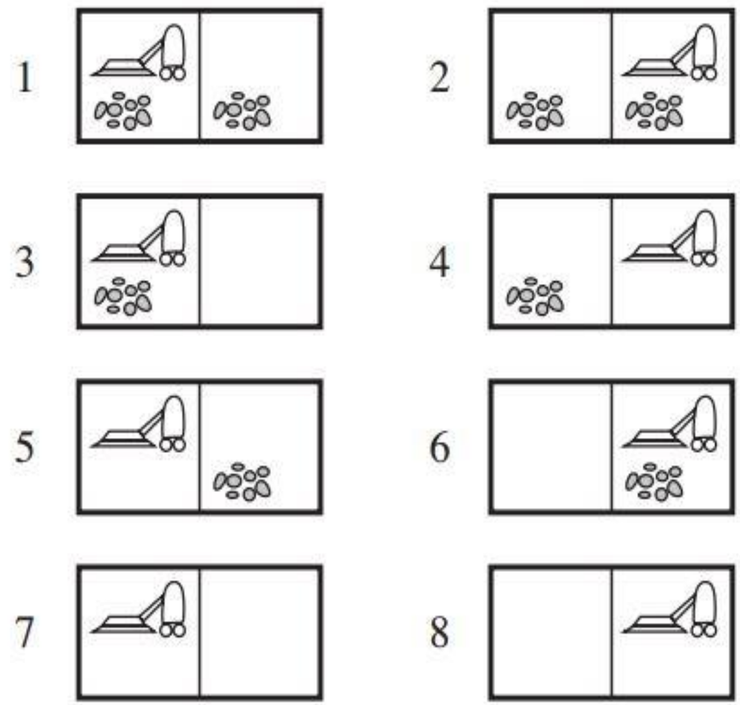
Single-state. Start in 5. Solution? (What's the goal?)



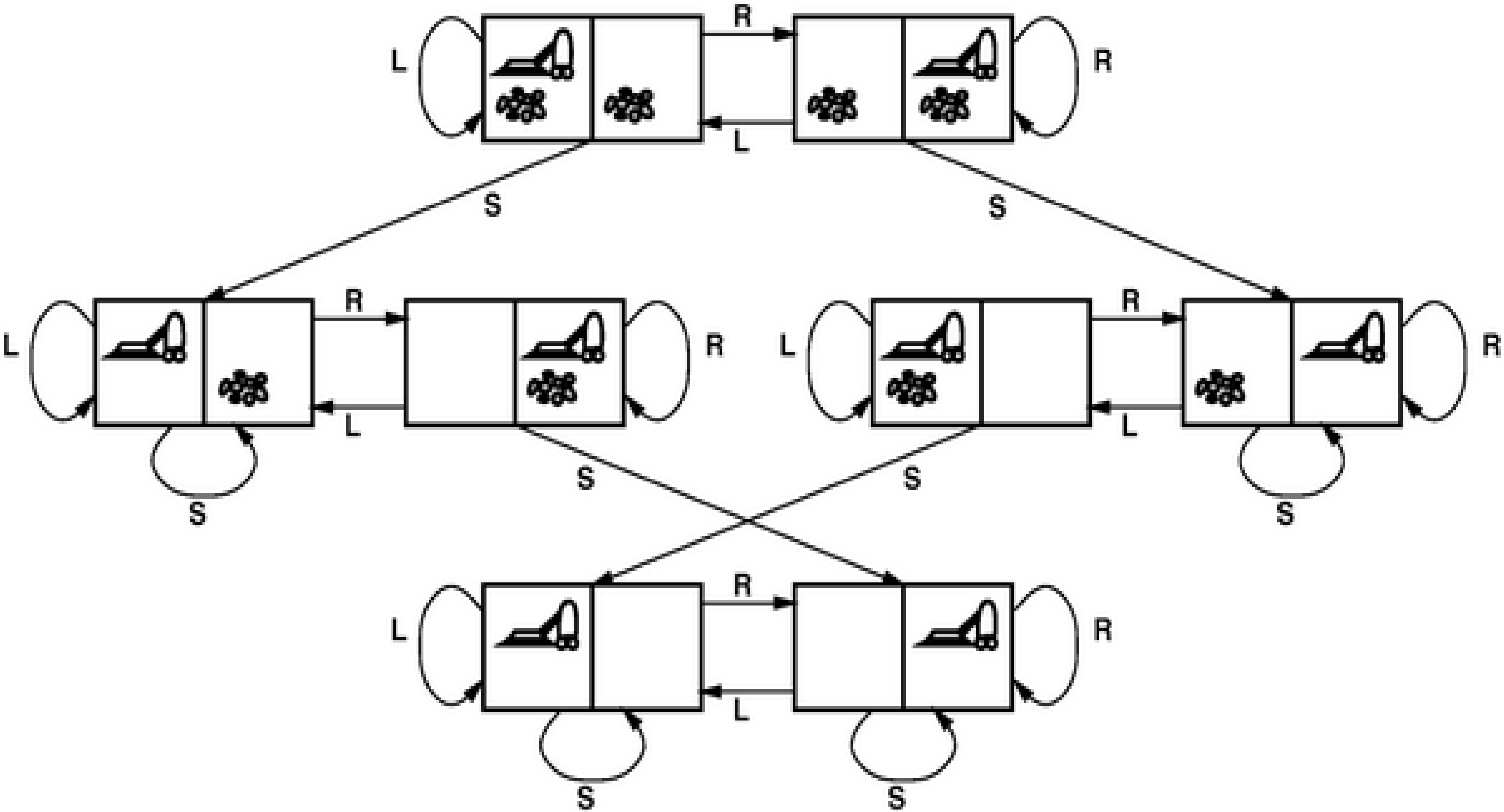
Example: Vacuum world

Single-state. Start in 5. Solution?

Solution: [Right, Suck]



Example: Vacuum World



Formulating a problem

Any lessons we can learn?

- States are abstractions of real-world configurations.
- Actions can be abstract but should be executable.
- Solutions should be feasible.
- Costs should be meaningful.