Artificial Intelligence
The Very Idea

Vasant G. Honavar
Dorothy Foehr Huck and J. Lloyd Huck Chair in Biomedical Data Sciences and Artificial Intelligence
Professor of Data Sciences, Informatics, Computer Science, Bioinformatics & Genomics and Neuroscience
Director, Artificial Intelligence Research Laboratory
Director, Center for Artificial Intelligence Foundations and Scientific Applications
Associate Director, Institute for Computational and Data Sciences
Pennsylvania State University

vhonavar@psu.edu
http://faculty.ist.psu.edu/vhonavar
http://ailab.ist.psu.edu

Agents and their environments

- We need to specify
  - Performance measure
  - Environment
  - Sensors
  - Actuators or Effectors

Environments

For a fully self-driving taxi:

- Performance
  - Safety, reaching destination, profits, comfort
- Environment
  - Streets, other traffic, pedestrians, weather, …
- Actuators
  - Steering, accelerating, brake, horn, speaker, display,…
- Sensors
  - Video, sonar, speedometer, engine sensors, keyboard, GPS, …

Environment types

- Observable versus partially observable
- Deterministic versus non-deterministic
- Episodic vs non-episodic
- Static vs dynamic
- Discrete versus continuous
- Single-agent versus Multiagent
- Open versus closed

Environment types

Fully vs. partially observable: an environment is fully observable when the sensors can detect all aspects that are relevant to the choice of action

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??				
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				
Open??				

Environment types

Fully vs. partially observable: an environment is fully observable when the sensors can detect all aspects that are relevant to the choice of action.

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				
Open??				

PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Environment types

Deterministic vs. stochastic: if the next environment state is completely determined by the current state the executed action then the environment is deterministic.

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				
Open??				

PennState College of Information Sciences And Technology — AI 100 Fall 2024 — Vasant G Honavar

Als de omgeving gedeeltelijk observeerbaar is kan deze stochastisch lijken.

Dus het is beter om te kijken of de omgebing determinitisch of stochastisch is vanuit het standpunt van de agent.

Als de omgeving deterministisch is behalve de acties van de andere agenten dan is de omgeving strategisch.

Environment types

Deterministic vs. non-deterministic: if the next environment state is completely determined by the current state and the action executed by the agent, then the environment is deterministic

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??				
Static??				
Discrete??				
Single-agent??				
Open??				

Environment types

Episodic vs. non episodic: In an episodic environment the agent's experience can be divided into episodes – agent's actions in consecutive episodes are independent

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??				
Static??				
Discrete??				
Single-agent??				
Open??				

Environment types

Episodic vs. non episodic: In an episodic environment agent's experience can be divided into episodes – agent's actions in consecutive episodes are independent

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	YES
Static??				
Discrete??				
Single-agent??				
Open??				

PennState Institute for Computational and Data Sciences
Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory
PennState Clinical and Translational Science Institute

Environment types

Static vs. dynamic: If the environment can change while the agent is choosing an action, the environment is dynamic

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	MAYBE
Static??				
Discrete??				
Single-agent??				
Open??				

PennState College of Information Sciences And Technology AI 100 Fall 2024 Vasant G Honavar

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Institute for Computational and Data Sciences | PennState Clinical and Translational Science Institute

Environment types

Static vs. dynamic: If the environment can change while the agent is choosing an action, the environment is dynamic

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	MAYBE
Static??	YES	YES	SEMI	NO
Discrete??				
Single-agent??				
Open??				

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Environment types

Discrete vs. continuous

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	MAYBE
Static??	YES	YES	SEMI	NO
Discrete??				
Single-agent??				
Open??				

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Environment types

Discrete vs. continuous

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	MAYBE
Static??	YES	YES	SEMI	NO
Discrete??	YES	YES	YES	NO
Single-agent??				
Open??				

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Environment types

Single vs. multi-agent: Does the environment contain other agents?

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	NO	MAYBE
Static??	YES	YES	SEMI	NO
Discrete??	YES	YES	YES	NO
Single-agent??				
Open??				

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Environment types

Single vs. multi-agent: Does the environment contain other agents?

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	YES
Static??	YES	YES	SEMI	NO
Discrete??	YES	YES	YES	NO
Single-agent??	YES	NO	YES (except auctions)	NO
Open??				

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Environment types

Open versus closed: Can entities enter and leave the environment?

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	MAYBE
Static??	YES	YES	SEMI	NO
Discrete??	YES	YES	YES	NO
Single-agent??	YES	NO	YES (except auctions)	NO
Open??				

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Environment types

Open versus closed: Can entities enter and leave the environment?

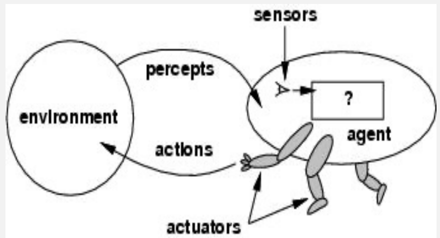
	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	FULL	FULL	PARTIAL	PARTIAL
Deterministic??	YES	NO	YES	NO
Episodic??	NO	NO	YES	YES
Static??	YES	YES	SEMI	MAYBE
Discrete??	YES	YES	YES	NO
Single-agent??	YES	NO	YES (except auctions)	NO
Open??	NO	NO	YES	YES

If the agent performance score changes ahen time passes, the environment is semi-dynamic.

Environment types

- The simplest environment is
  - Fully observable, deterministic, episodic, static, discrete and single-agent
- Many real world environments can be:
  - Partially observable, stochastic, non-episodic, dynamic, continuous and multi-agent

Agent types



All agents have:

- Input = current percept
- Output = action
- Program = processes input to produce output

Types of Agents

Function TABLE-DRIVEN_AGENT(*percept*) **returns** an action

maintain: *percepts*, a sequence initially empty
table, a table of actions, indexed by percept sequence

append *percept* to the end of *percepts*
action ← LOOKUP(*percepts*, *table*)
return *action*

This approach is doomed to failure in all but the simplest cases
Why?

Agent types

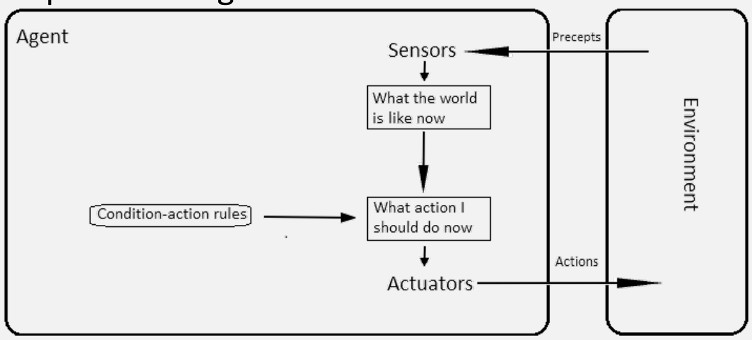
Basic agent types

- Simple reflex agents or reactive agents
- Model-based reflex agents
- Deliberative agents (often goal-based)
- Utility-based agents

All these agent types can be

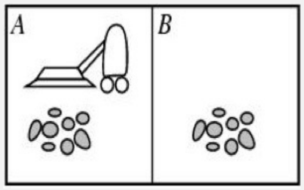
- Learning
- Communicative
- Interactive (competitive, collaborative)

Simple Reflex Agent



- Select action on the basis of *only the current* percept
  - E.g. the vacuum-cleaner agent
- Implemented through *condition-action rules*
  - Example: vacuum cleaner agent
  - If dirty then cleanup

The vacuum-cleaner agent



Agent function

if *status is Dirty* then *Cleanup*

else if *location is A* then move *Right*

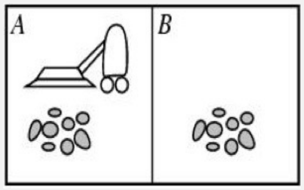
else if *location is B* then move *Left*

- Note that this design works only for the room layout shown
- A more general design needs to use a map of the floor and plan to choose its moves

Simple reflex agents

- Reflex agent
  - Has limited intelligence
  - Has only a limited view of the state of the environment
  - Has program specified by the designer, and modified if the environment in which it needs to function changes
  - design can be generalized to work in different environments simply by specifying a set of condition-action rules to be interpreted and applied in different contexts.

The vacuum-cleaner agent



Agent function

if *status is Dirty* then *Cleanup*

else if *location is A* then move *Right*

else if *location is B* then move *Left*

- Suppose its location sensor fails and it is forced to operate with only its dirt sensor.
- Now it has only two possible percepts, namely, status being *Dirty* or *Clean*.
- Now if the status is *Dirty*, it will activate the first rule, triggering the action Cleanup.
- But what about if the status is Clean?

Reflex agents

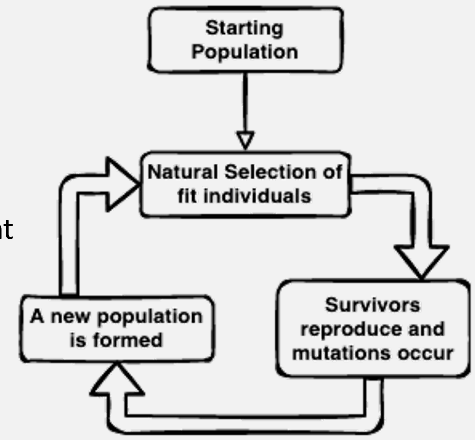
- Because of the failure of its location sensor, the has no information about its current location.
- Consequently, none of the rules match its current percept Clean, thereby effectively paralyzing the vacuum cleaner agent.
- In this case, one can “fix” this problem by adding a random aspect to the agent function, that is, when no rules match the current percept, randomly move to the other square.
- In general, the inability to have a complete picture of the state of the environment makes the agent’s task harder, and in some cases, impossible.

Simple reflex agents

- Given a suitable agent function, a reflex agent succeeds only when the environment is fully observable.
- When the environment is only partially observable, it has no way to distinguish between two or more distinct environmental states that are indistinguishable from the agent's point of view.
- Because the agent cannot distinguish between the actual environmental states

Simple reflex agents and Dennett's Darwinian creatures

- Generate a population of agents with random agent functions
- Repeat:
  - Assess the fitness of agents in the environment
  - Use simulated evolution to perform fitness proportionate selection and recombination to produce the next population



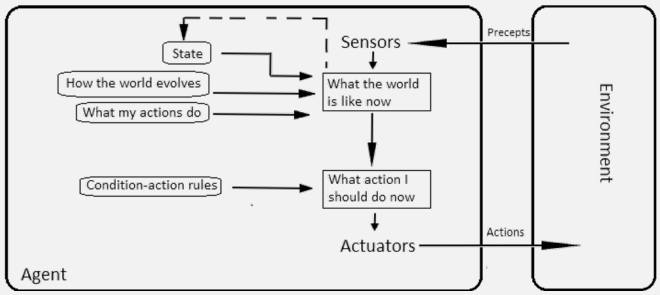
From simple reflex agents to reflex agents with state

- Consider a mars rover agent designed to pick up rock samples of distinct types on the Mars surface.
- A reflex agent would be able to pick up every rock it encounters (if *you see a rock*, then *pick it up*)
- It will not be able to ensure that each rock sample it collects is of a different type than those that it has already collected
- Why not?
- It does not and it cannot keep track of the types of rocks it has collected.

From simple reflex agents to reflex agents with state

- A reflex agent with state can encode as part of its internal state description, the types of rocks it has collected already
- It can decide whether to pick up the current rock sample based on whether it has already collected a rock of the same type or not

Reflex agents with state



- Maintains internal state
- Over time update state using based on its knowledge and observations of the world
  - How does the world change?
  - How do the actions affect the world?
- Can handle partially observable environments

Reflex agents with state

- Find a rule that matches current state and percept
- Apply the rule to choose action
- Perform action
- Update state

Reflex agents with state

- State update requires
  - a state transition model that describes how the world changes over time
    - due to its intrinsic dynamics, independently of the agent's actions and
    - due to the agent's actions.
  - A sensor model that describes how aspects of state of the world are mapped to percepts.

- Example: in the case of the vacuum cleaner agent, the status of a location changes from *Dirty* to *Clean* when the Cleanup action is executed by the agent at that location

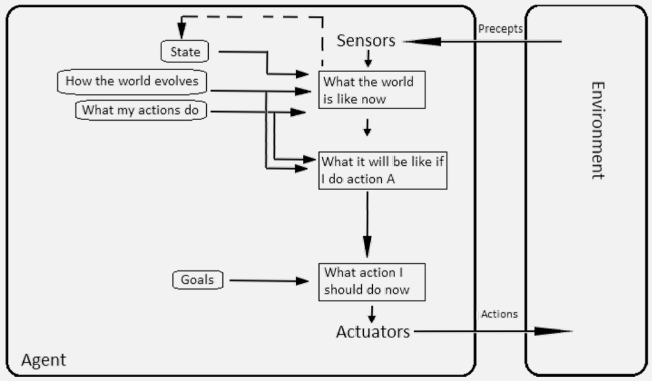
Agents with goals

- Even perfect of the current state of the environment is not necessarily sufficient for an agent to decide what it should do
- Consider a self-driving taxi at a four-way intersection in Washington DC.
- The taxi has three choices: turn left, turn right, or go straight.
- The right thing to do depends on not just the state of the environment, but where the taxi needs to get to
- Without a destination to get to, all three actions are equally good so the the taxi would drive around aimlessly

Agents with goals

- Suppose the taxi has a goal that it needs to achieve, for example, get to the Union Station in Washington DC
- The taxi's decision as to what to do next can be informed by its goal of getting to the Union Station
- Example: Taxi should turn right if that puts it on the road that takes it directly to the Union Station as opposed to turning left which leads it away from the Union Station

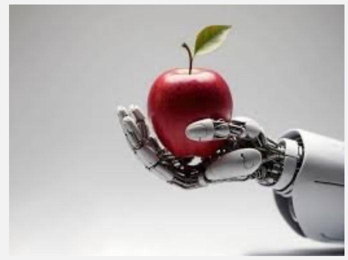
Goal-based agents



- Agent is provided a goal
- The agent seeks to achieve the specified goal
- Attaining a goal may require a long sequence of actions
- Needs a model (representation) of the world

Goal-based agents

- In some scenarios, selecting an optimal action based on the goal is straightforward.
- Example:
  - If the goal is to get an apple, and
  - the apple is within reach of the agent,
  - and one of the possible action choices is pickup the apple,
  - the goal can be achieved by simply by executing that action

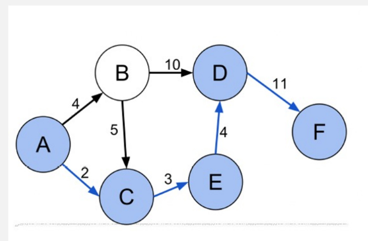


Goal-based agents

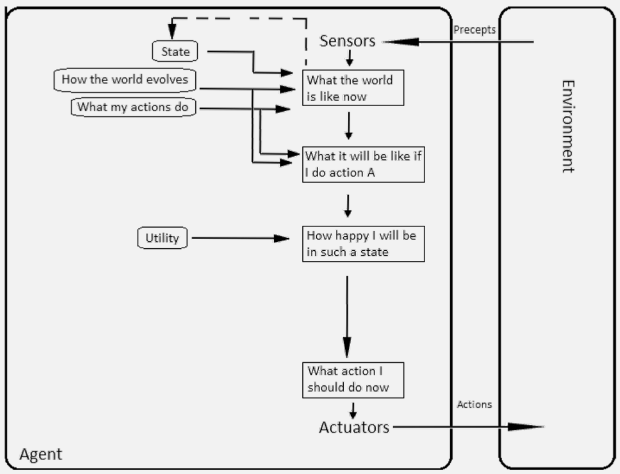
- In other scenarios, achieving a goal may need an entire plan or a sequence of actions
- Example:
  - Get in the car → Drive to the grocery store → Park the car → Get out of the car → Walk into the grocery store → Find the aisle with fruits → Find apple → Pickup apple → Pay → Walk out of the store → Get in the car → Drive home → Park the car → Enter the home
- Finding such a plan requires knowledge of
  - The preconditions and effects of actions
    - In order for the agent to drive the car, the agent must be seated in the car, have car keys, and the license to drive, and so on.
  - Some knowledge of the world e.g., that grocery stores sell apples …

Optimal plan seeking agents

- A goal-based self-driving taxi equipped with the appropriate knowledge base and planning abilities can take you from your current location, say Ronald Reagan International Airport, to the Union Station in Washington DC.
- But it may take an inordinately long route, or it may take an extremely busy route, when there are better options available.
- We may want the agent to be able to find, given some notion of optimality, an optimal plan instead of simply finding any plan that would take you from the start state to the goal state.



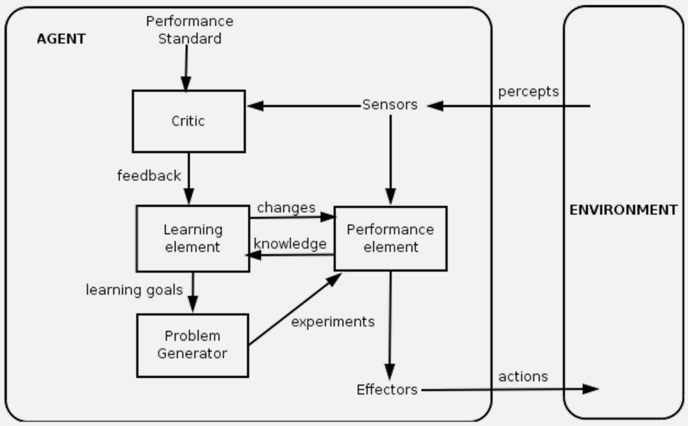
Utilitarian agent



- Different goals have different utilities for the agent
- Utility function maps a (sequence of) state(s) onto a real number (utility)
- Utility can influence choice of action
- Goal-based agents are a special case (utility is 1 for the goal state, and 0 for all other states)

- How do agents come to possess the knowledge that they need?
- The agent may be programmed by their designers through a laborious process of encoding all the relevant bits of knowledge about the world that the agent would need to satisfactorily perform tasks that it is expected to perform in its environment
- Is there a better way?
  - Learning from their interactions with the environment, other agents, including humans.
  - Learning from examples, environmental reward/punishment, advice, imitation, active demonstration, among other means

Learning agents



- Agents use knowledge (e.g., model of the environment)
- Where does this knowledge come from? – Learning

Learning agents

- Learning element improves performance
- Performance element realizes agent programs
- Problem generator suggests actions leading to informative experiences

Learning agents

- One of the simplest forms of learning is Skinnerian conditioning
  - A reflex agent randomly tries out different actions from its repertoire that match the percept
  - The environment provides reward / punishment
  - Eventually the agent figures out the optimal action to perform for each percept
- A major limitations of learning agents fashioned after Skinnerian creatures is that their success depends on being lucky enough to try out actions that are not dangerously bad.
- Example:
  - A self-driving car that randomly tries out stepping on the accelerator as it approaches a red light.
  - The car will crash and get destroyed with no chance to learn from its experience.
- Agents must start with agent programs that allow them to limit their choice of actions to those that are not downright dangerous

Learning Agents

- How can an agent limit its choice of actions to those that are better than chance?
  - The agents start out with some basic knowledge of the world and the ability to imagine and reason about the effects of their actions.
    - Example: The vacuum cleaner agent could know by design that it is dangerous to go into a flooded room because of the risk of dangerous short-circuit.
  - The agents could use virtual worlds for learning to limit their action choices
    - Driving simulator for training self-driving taxis
    - Flight simulator

Multi-agent Systems

- Inter-agent communication
- Multi-agent Interaction
  - Cooperation
  - Competition
- Multi-agent organizations