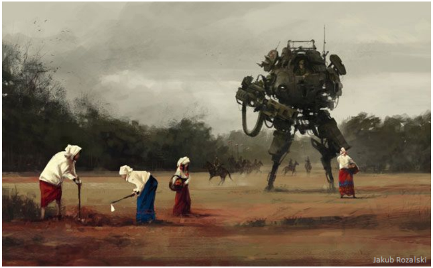
CS761 Artificial Intelligence

Agents and World Models

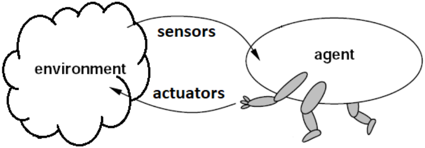
Towards Intelligent Agents

- A computational agent is an entity capable of
  - perceiving the environment;
  - deliberating;
  - performing actions
- Artificial Intelligence can be viewed as the study of the synthesis and analysis of computational agents that act intelligently.



An agent task is defined by:

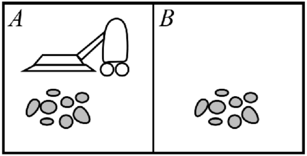
- Environment: The states of the task
- Sensors: Representations of information about the environment states that are input to the agent system
- Actuators: Actions that may modify the environment states
- Performance measure: Evaluation function of any given sequence of environment states



Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

- Percept is the agent's perceptual input at any given instant.
- An agent's percept sequence is the complete history of everything that the agent has perceived.
- An agent program maps a given percept sequence to an action. This function describes the agent's behaviours.

Example [vaccum world].



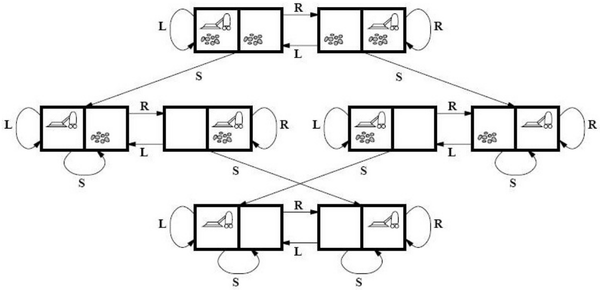
Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$

Agents with a World Model

- World model: Encodes information about
  - How the world evolves independently of the agent;
  - How the agent's actions affect the world.
- A model-based agent program uses a world model to determine the agent's actions. In particular,
  - The agent maintains an internal state in the world model;
  - The internal state is updated by a state-transition function based on the agent's percept history;
  - (Optional) An agent's behaviours may be directed by goals or utilities.[1]

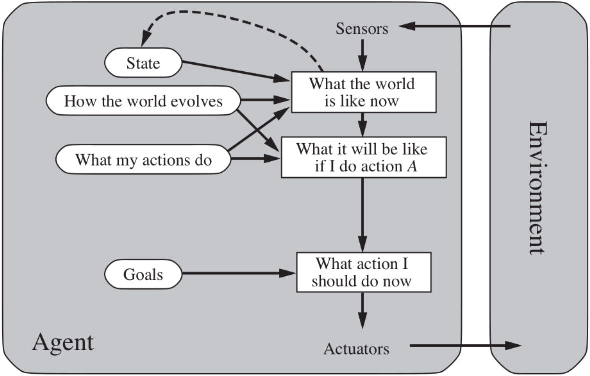
[1] To be discussed in future lectures

Example. The following is a state-based world model of the vacuum world scenario.



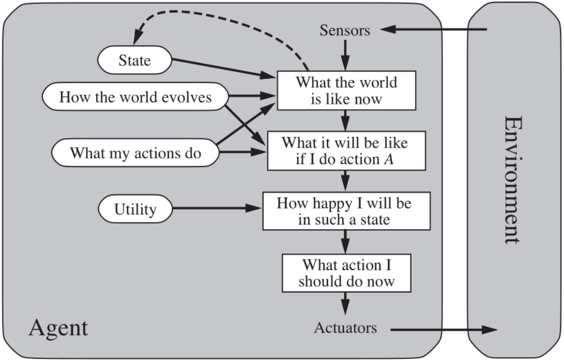
Example. [Goal-based agents] A goal-based agent program maintains explicit information about the situations that are desired by the agent.

E.g., In the vacuum world scenario, the goal could be to achieve a state where all locations are clean.



Example. [Utility-based agent] A utility-based agent program uses a utility function to measure how desirable a state is, and actions are derived to maximise utility.

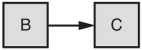
E.g., In the vacuum world scenario, a utility function may be the number of dirty locations cleaned.



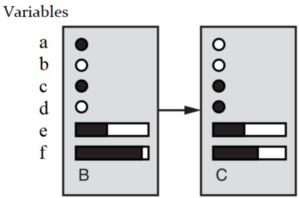
Challenge: Choose an explicit representation of a world model.

So far we have seen two representations of world models:

1. Atomic representation: Explicitly represent each state of the world and their transitions.
    - States are atomic entities in the model.
    - Any action leads to a state transition (from one state to another).
2. Factored representation: A state of the world is captured by a collection of variables and their values.
    - States are variable assignments that assign every variable to a value in its domain.
    - Any action leads to some potential changes to the variable values, thereby state changes.

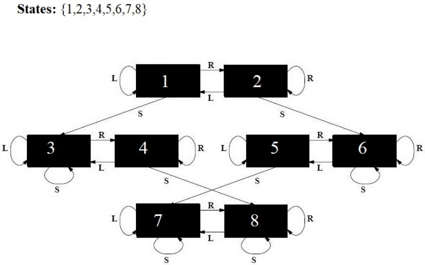


(a) Atomic

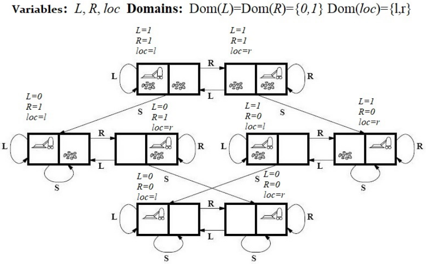


(b) Factored

Example. The vacuum world scenario may have the following atomic and factored representations:



Atomic Representation



Factored Representation

Structured Representation

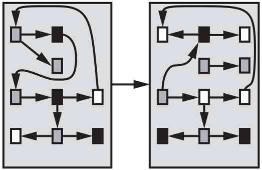
Simply keeping the variable values is often insufficient:

- An agent may only have partial knowledge about the state;
- The variables may correlate with each other through constraints, thus the potential for reasoning. E.g. Being a fish & Living under water.

Structured representation

A structured representation is world model that describes the world using variables, while capturing the following:

- Knowledge: Constraints on variables.
- Reasoning: Inference engines to derive knowledge from percepts.



Agent with a Knowledge Base

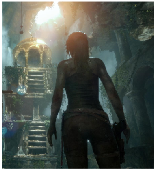
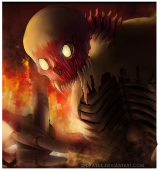
In the following, we look at a scenario where an agent would need to apply knowledge about the world, and reason about states.

Example. [Wumpus world] An agent enters a tomb.

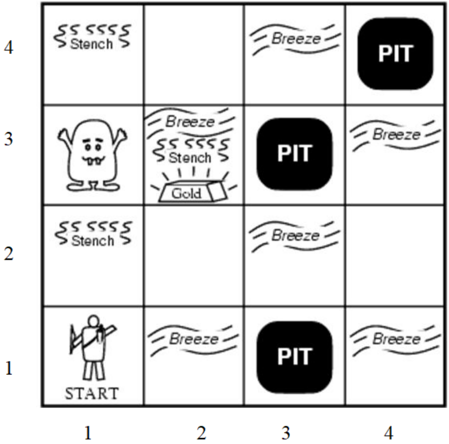
- A wumpus is a monster that eats the agent if the agent walks in the same cell.
- A pit is a trap that kills the agent if the agent steps on it.
- A gold is a treasure that the agent wants to get.

The agent can perceive information about her current location:

- The wumpus is stink, and the agent can sense stench when walking beside it.
- Wind blows out of the pit, and the agent can sense breeze when walking beside a pit.
- The gold glitters, so the agent can see it once the agent steps on it.

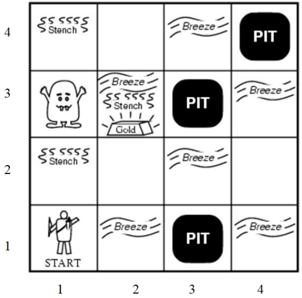
For simplicity, we abstract the game into a 4×4 grid:



Note: The map above is only known by the game designer, but not the agent when she starts.

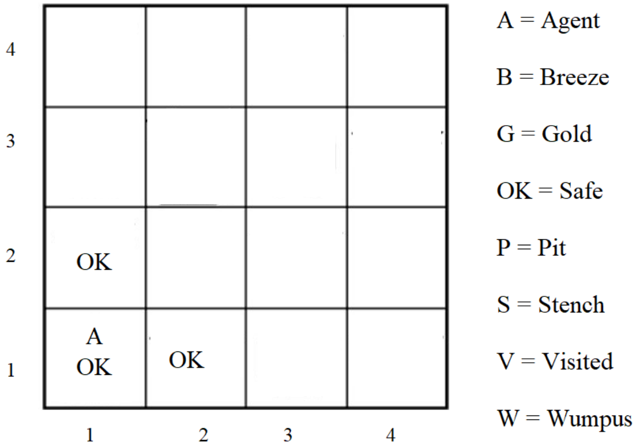
The agent's knowledge, i.e., constraints:

- A location (i, j) can either be safe or unsafe (in the presence of wumpus or pit).
- If a wumpus is at (i, j), then $(i-1, j)$, $(i, j-1)$, $(i+1, j)$, and $(i, j+1)$ have stench (if these cells exist).
- if a pit is at (i, j), then $(i-1, j)$, $(i, j-1)$, $(i+1, j)$, and $(i, j+1)$ have breeze (if these cells exist).
- If a location (i, j) glitters, then we find gold.



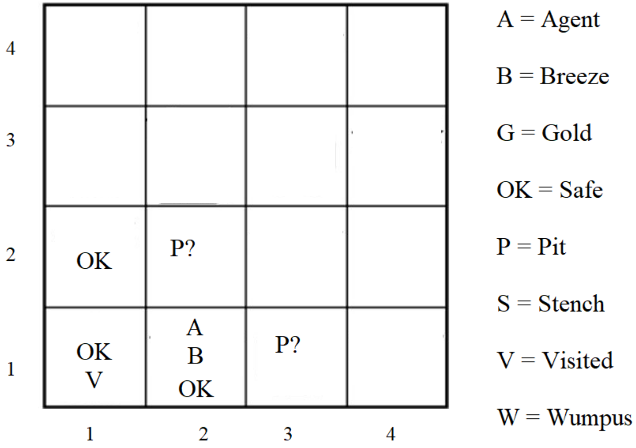
Step 0: The agent starts at $(1,1)$.

- Percept: $(1,1)$ is safe.
- Inference: There is no B nor S at $(1,1)$, so $(1,2)$, $(2,1)$ are safe.



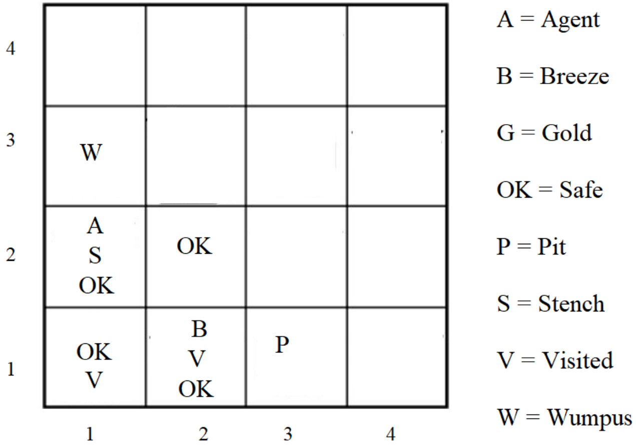
Step 1: The agent moves right to $(2, 1)$.

- Percept: $(2, 1)$ has B.
- Inference:
  - Either $(2, 2)$ or $(3, 1)$ has a P, or both have.
  - $(2, 2)$ and $(3, 1)$ do not contain W.



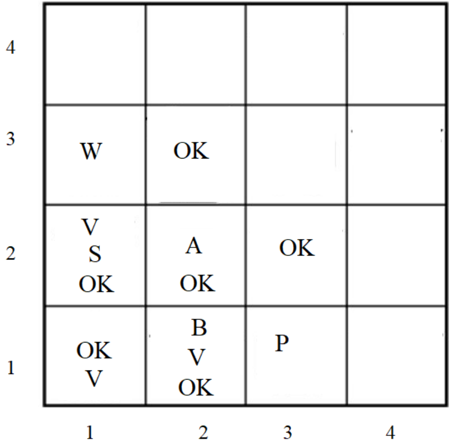
Step 2: The agent moves to $(1,2)$.

- Percept: $(1,2)$ has S.
- Inference:
  - $(2,2)$ has no P. Thus is safe.
  - $(1,3)$ must has W.



Step 3: The agent moves to $(2,2)$.

- Percept: $(2,2)$ has no B nor S.
- Inference: $(2,3)$ and $(3,2)$ are both safe.



A = Agent

B = Breeze

G = Gold

OK = Safe

P = Pit

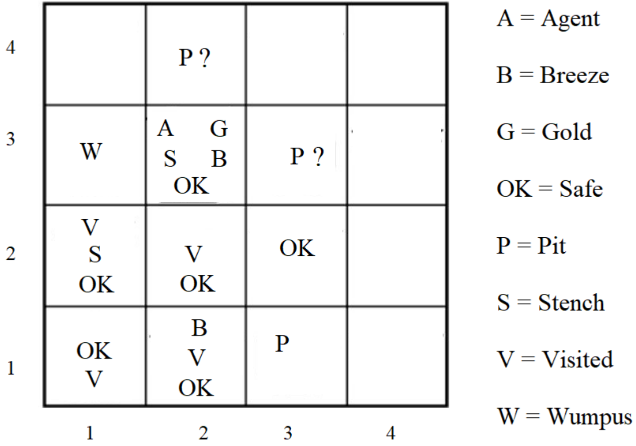
S = Stench

V = Visited

W = Wumpus

Step 4: The agent moves to $(2, 3)$.

- Percept: $(2, 3)$ has B, S and G.



Knowledge-based Agents

The wumpus world scenario needs a structured representation to code constraints and percepts, and to perform inferences.

Knowledge-based Agents

The wumpus world scenario needs a structured representation to code constraints and percepts, and to perform inferences.

The agent consists of two parts:

- **Offline:** the agent has a knowledge base (KB), which stores previously held constraints.

  E.g. "A wumpus is stink and all locations around it will have stench"

  "A pit has air through it so all locations around it will have breeze"

- **Online:** the agent implements an inference engine (IE), i.e., algorithms that generate new constraints from percepts and KB.

  E.g. "Location $(1, 1)$ has no breeze nor stench. Thus $(1, 2)$ and $(2, 1)$ are both safe".

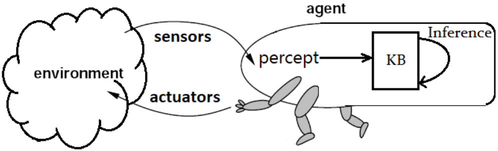
  "Given $(1, 1)$ has no wumpus, since $(2, 1)$ has no stench but $(1, 2)$ has stench, $(1, 3)$ must has wumpus".

Knowledge-based Agent

A knowledge-based agent implements the following:

- A knowledge base is a set of constraints.
- A knowledge base can be updated through an inference engine.

A knowledge representation language must be used to specify constraints in a knowledge base.



Summary of The Topic

The following are the main knowledge points covered:

- **Agent tasks:** Environment, Sensors, Actuators, Performance measure
- **Agent program:** Mapping percept sequences to actions.
- **Model-based agents:**
  - States
  - State transitions
  - Goals or utilities
- **Structured representation:** Knowledge, reasoning
- **Knowledge-based agent:**
  - Knowledge representation language: Specify a structure representation of a world model.
  - Knowledge base (KB): facts specified by a knowledge representation language.
  - Inference engine (IE): algorithm that generate new facts from percepts and KB.