# Lecture 11
# Logical Agents, Knowledge Representation & Reasoning: Part I

Rob Gaizauskas

# Lecture Outline

- Knowledge-based Agents
- Wumpus World
- Logic
  - Syntax
  - Semantics
  - Model Checking
  - Inference
- Agents Based on Propositional Logic
- Reading: (Readings that begin with * are **mandatory**)
  - *Russell and Norvig (2010), Chapter 7 “Logical Agents”, sections: 7.1-7.4 and 7.7

# Lecture Outline

- Knowledge-based Agents ← Today
- Wumpus World
- Logic
  - Syntax
  - Semantics
  - Model Checking
  - Inference
- Agents Based on Propositional Logic
- Reading: (Readings that begin with * are **mandatory**)
  - *Russell and Norvig (2010), Chapter 7 "Logical Agents", sections: 7.1-7.4 and 7.7

# Knowledge-based Agents

- Humans know things and can reason using this knowledge to help them do things
  - E.g. if I know Fred is in the next room and has the key to the lab then I can reason that to get into the lab I should go and ask Fred for the key
- I.e. humans do not act purely by reflex but by processes of reasoning that work over internal representations of knowledge
  - In AI this approach to intelligence – modelling internal representations of knowledge and processes of reasoning – is embodied in knowledge-based agents

# Knowledge-based Agents

- Central component of a knowledge-based agent is its knowledge base or KB
- A KB is a set of sentences
  - A sentence is an assertion about the world
  - Sentences are expressed in a knowledge representation language
  - A sentence that is not derived from other sentences is sometimes called an axiom

# Knowledge-based Agents and Inference

- Need to be able to add sentences to the KB and query the KB to find out what is known
  - These operations are usually called TELL and ASK
- Both TELL and ASK may involve inference – deriving new sentences from old
  - Must be such that when ASKing the KB a question the answer must follow from what the KB has been told (TELLed)

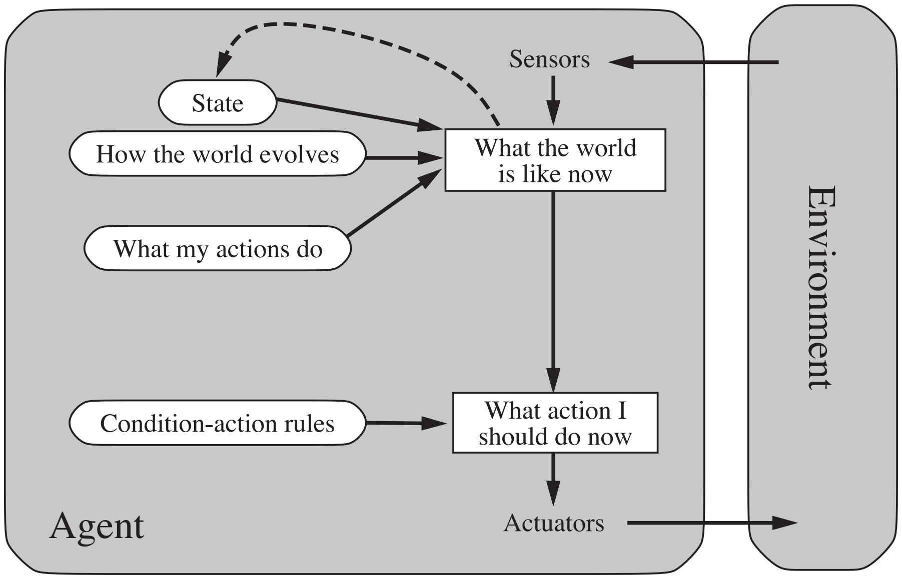
# A Knowledge-based Agent Program

**function** KB-AGENT(*percept*) **returns** an *action*
  **persistent**: *KB*, a knowledge base
      *t*, a counter, initially 0, indicating time

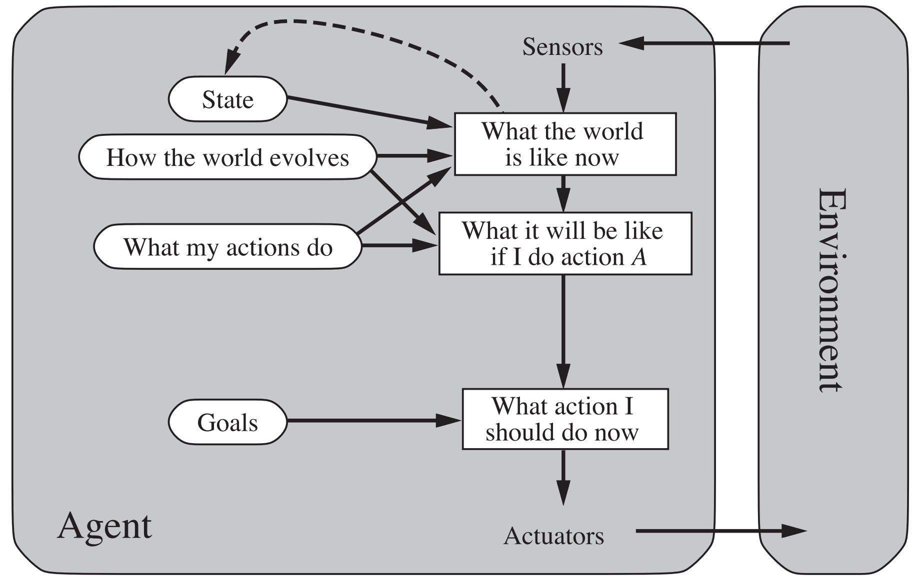
  TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))
  *action* ← ASK(*KB*, MAKE-ACTION-QUERY(*t*))
  TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))
  $t \leftarrow t + 1$
  **return** *action*

- Program takes percept as input and returns an action
- Agent maintains a KB – may initially contain some background knowledge
- Each time agent is called it:
  1. TELLs the KB what it has perceived
     - May involve reasoning as part of interpreting raw percepts as, e.g. an object type in a scene type
  2. ASKs the KB what action it should perform
     - May involve reasoning about state of the world + outcomes of possible action sequences
  3. TELLs the KB which action was chosen
- Agent then executes the action

# KB Agents and Agent Structure




Model-based Reflex Agent (lecture 8)




Goal-based Agent (lecture 8)

- Recall agent structures discussed in Lecture 8
- A KB agent is one form a model-based reflex agent, goal-based agent or a utility-based agent could take

# Knowledge vs Implementation Level

- KB-Agent is like agents with internal state introduced in Lecture 8
- But, because of definitions of TELL and ASK, it is not an arbitrary program for calculating actions
  - Program can be described at the knowledge level
  - I.e. only need to specify what the agent knows and what its goals are to fix its behaviour
- Example:
  - Robot has goal of going from DCS Reception to the Lewin Lab
  - Knows lift is the only way for it to get from Level 1 to Level 0
  - Then we can expect it to take lift *because it knows that will achieve its goal*
- This analysis independent of how robot works at implementation level
  - Knowledge of building layout may be implemented as lists or hash tables
  - Reasoning may be carried out by manipulating strings of symbols or propagating noisy signals in a neural network

# Knowledge vs Implementation Level

- KB-Agent is like agents with internal state introduced in Lecture 8
- But, because of definitions of TELL and ASK, it is not an arbitrary program for calculating actions
  - Program can be described at the knowledge level
  - I.e. only need to specify what the agent knows and what its goals are to fix its behaviour
- Example:
  - Robot has goal of going from DCS Reception to the Lewin Lab
  - Knows lift is the only way for it to get from Level 1 to Level 0
  - Then we can expect it to take lift *because it knows that will achieve its goal*
- This analysis independent of how robot works at implementation level
  - Knowledge of building layout may be implemented as lists or hash tables
  - Reasoning may be carried out by manipulating strings of symbols or propagating noisy signals in a neural network

Doesn't matter

# Declarative vs Procedural Approaches to System Building

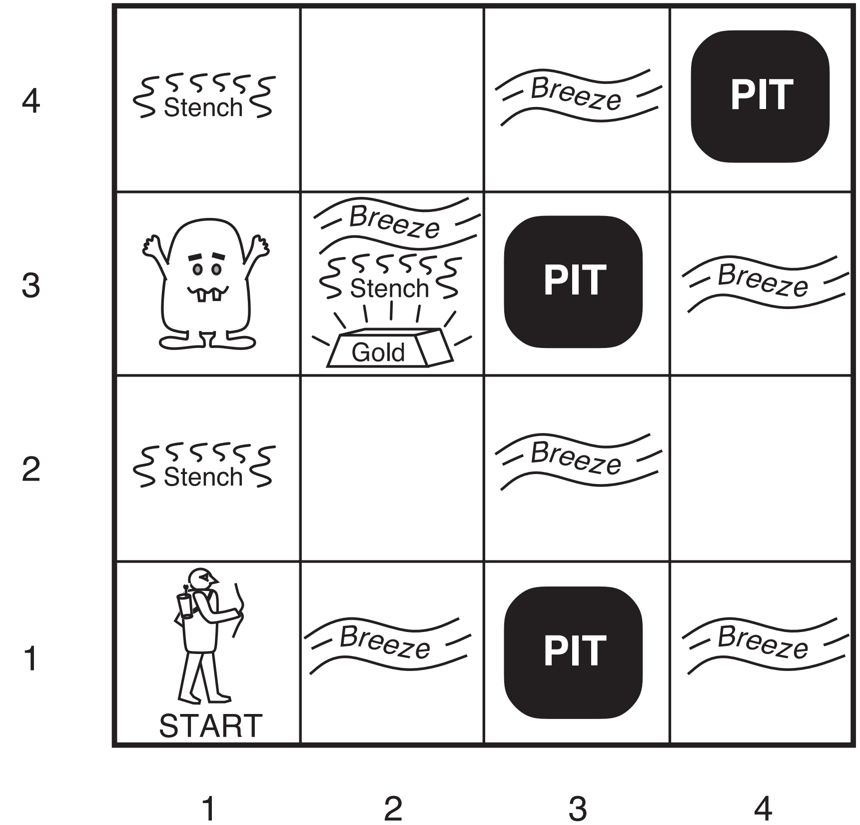
- A KB agent can be built by TELLing it what it needs to know
  - Start with an empty KB
  - Designer TELLs it sentences 1-by-1 till it knows how to operate in its environment
  - Called the declarative approach to system building
- Alternative:
  - Encode desired behaviour directly as program code
  - Called the procedural approach to system building
- 1970s and 80s saw heated debates between proponents of two approaches
  - Now realize both approaches useful & that declarative knowledge can often be compiled into more efficient procedural code
- Note: can provide a KB-agent with mechanisms for learning
  - Can create general knowledge about environment from series of percepts
  - Such agents can be fully autonomous

# Wumpus World: Extended Example

- The power of the knowledge-based agent approach can be demonstrated via a simple game in an imaginary environment called the wumpus world
- The wumpus world:
  - is a cave consisting of rooms connected by passageways
  - wumpus is a beast – lurks in a room in the cave and eats anyone who enters its room
  - wumpus can be shot by agent; but agent has only 1 arrow
  - some rooms have bottomless pits
  - somewhere in the cave is a heap of gold
- Simple by modern game standards
  - But good illustration of important points about intelligence

# PEAS description of Wumpus World

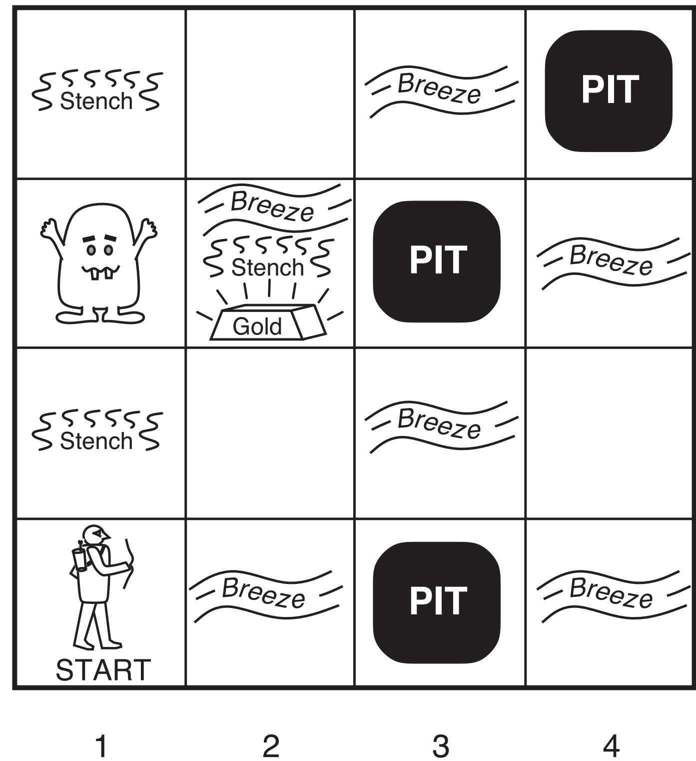
- Task environment specified by PEAS (lecture 7) description:
- Performance measure:
  - +1000 for climbing out of cave with gold
  - -1000 for falling into a pit or being eaten by the wumpus
  - -1 for each action taken
  - -10 for using up the arrow
  - Game ends when agent dies or exits the cave successfully




- Environment:
  - A 4 x 4 grid of rooms
  - Agent always starts in [1,1] facing right
  - Position of gold and wumpus chosen randomly with uniform distribution from squares other than [1,1]
  - Each square other than [1,1] can be a pit with probability 0.2

# PEAS description of Wumpus World

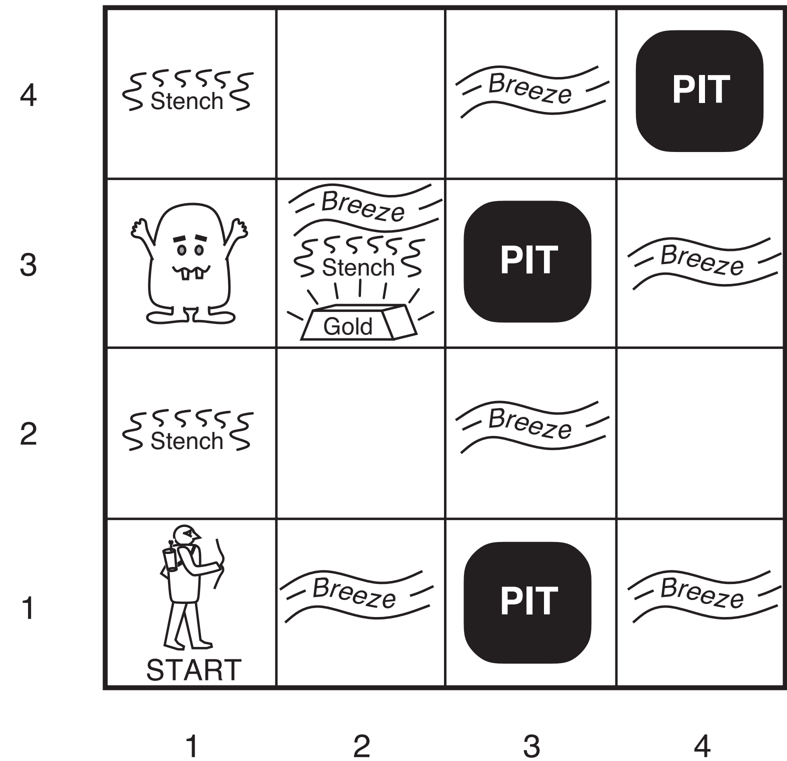
- Actuators:
  - Agent can move Forward, TurnLeft by 90° or TurnRight by 90°
  - Agent dies if it enters a room with a pit or a live wumpus
  - If agent tries to move forward and bumps into wall then does not move
  - Action Grab picks up gold, if agent is in same room as gold
  - Action Shoot fires an arrow in straight line in direction of agent – continues until it either  kills wumpus or hits wall (only 1 arrow)
  - Action Climb allows agent to exit cave from [1,1]



# PEAS description of Wumpus World

- Sensors: agent has 5 sensors:
  - Agent perceives a Stench in a room directly adjacent to the wumpus
  - Agent perceives a Breeze in a room directly adjacent to a pit
  - Agent perceives a Glitter in the room with the gold
  - Agent perceives a Bump when it walks into a wall
  - Wumpus when killed, emits a Scream perceivable anywhere in the cave

Percepts given to agent as a quintuple, e.g. if there is a stench and a breeze but no glitter, bump or scream then agent gets: [Stench, Breeze, None, None, None]



# Characteristics of Wumpus Environment

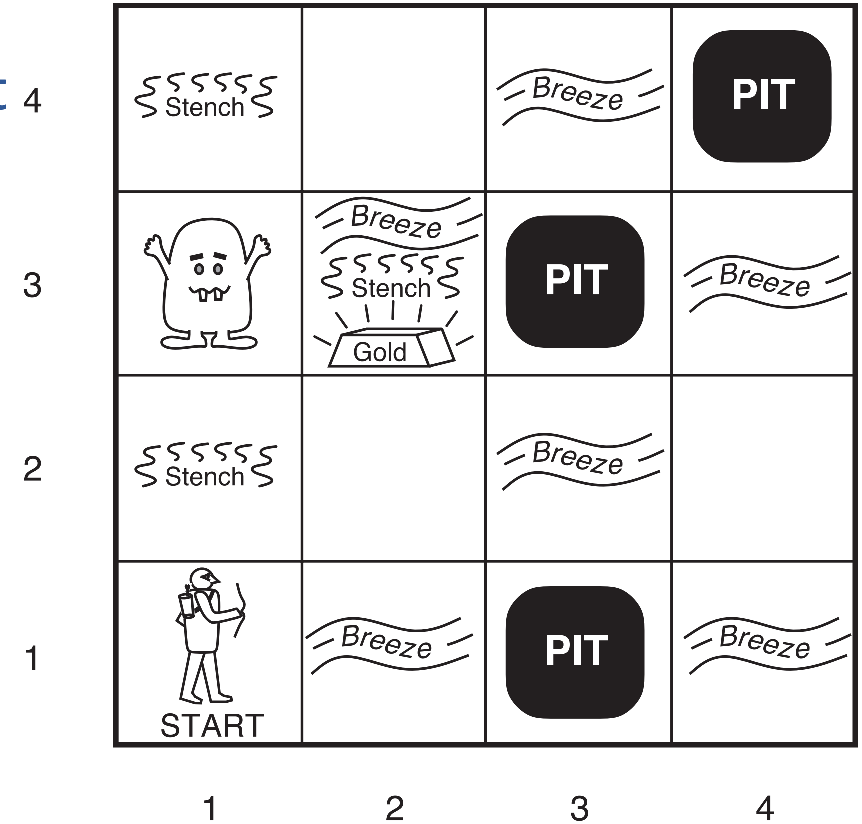
- In terms of distinctions introduced in Lecture 7 environment is:
  - Discrete: finite number of discrete states, time handled discretely, agent has finite number of distinct percepts and actions
  - Static: environment does not change while agent is deliberating
  - Sequential: the current decision could affect all future decisions
  - Partially observable: some aspects of state not directly perceivable
    - Agent location, wumpus's state of health; availability of arrow
    - Location of pits and wumpus
      - Could treat as unobserved parts of state – in which case transition model for environment completely known
      - Or treat transition model as unknown (which Forward actions are fatal?) – so discovering their locations completes knowledge of transition model
  - Deterministic: next state of the environment is completely determined by the current state and the agent's action
  - Single-agent

# Characteristics of Wumpus Environment

- Main challenge for agent is initial ignorance of configuration of environment
  - Overcoming this ignorance appears to require logical reasoning
- In most instances of the world agent can retrieve gold safely
  - Sometimes agent must choose between going home without gold or risking death to retrieve it
  - About 21% of environments are unfair as gold is either in a pit or surrounded by pits

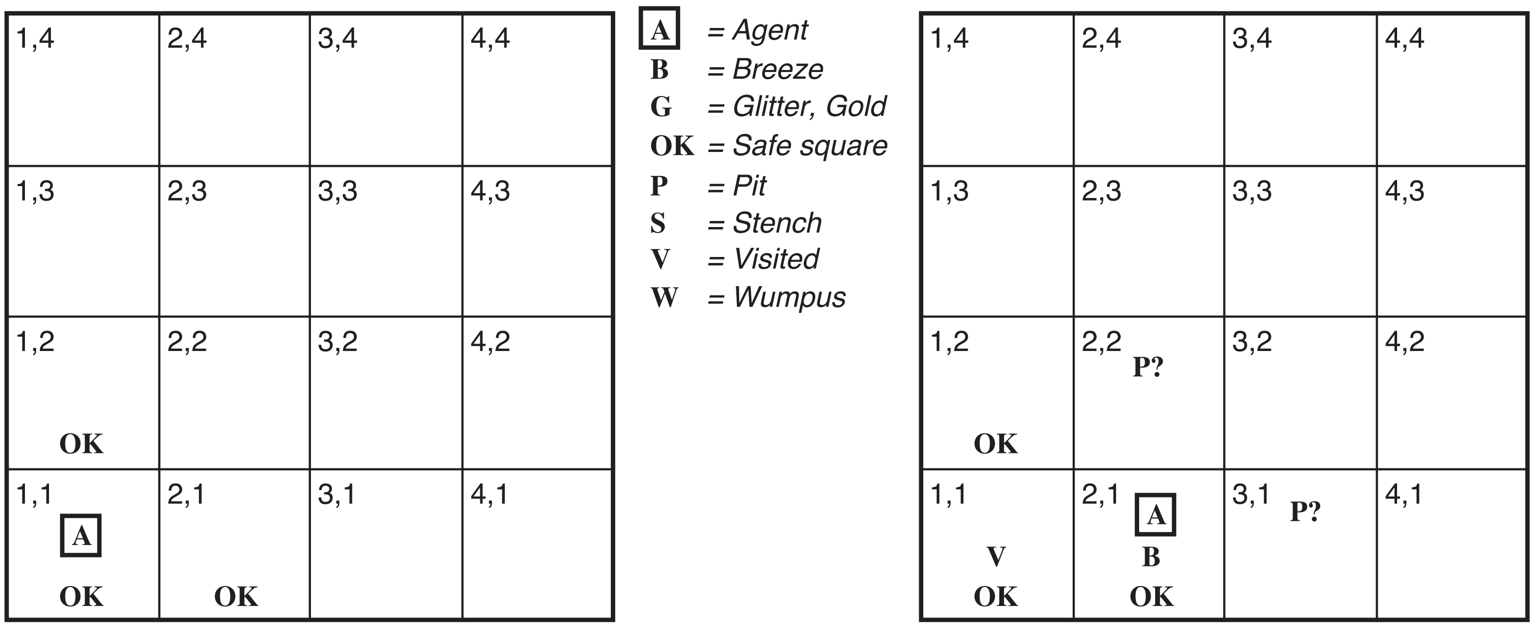
# Exploring in the Wumpus World

- Consider a knowledge-based agent exploring this wumpus world
- Agent's initial KB contains rules of the environment as described above:
  - Knows it starts in [1,1] and that this is a safe square
- First percept is [*None,None,None,None,None*]




- From this agent can conclude that [1,2] and [2,1] are safe

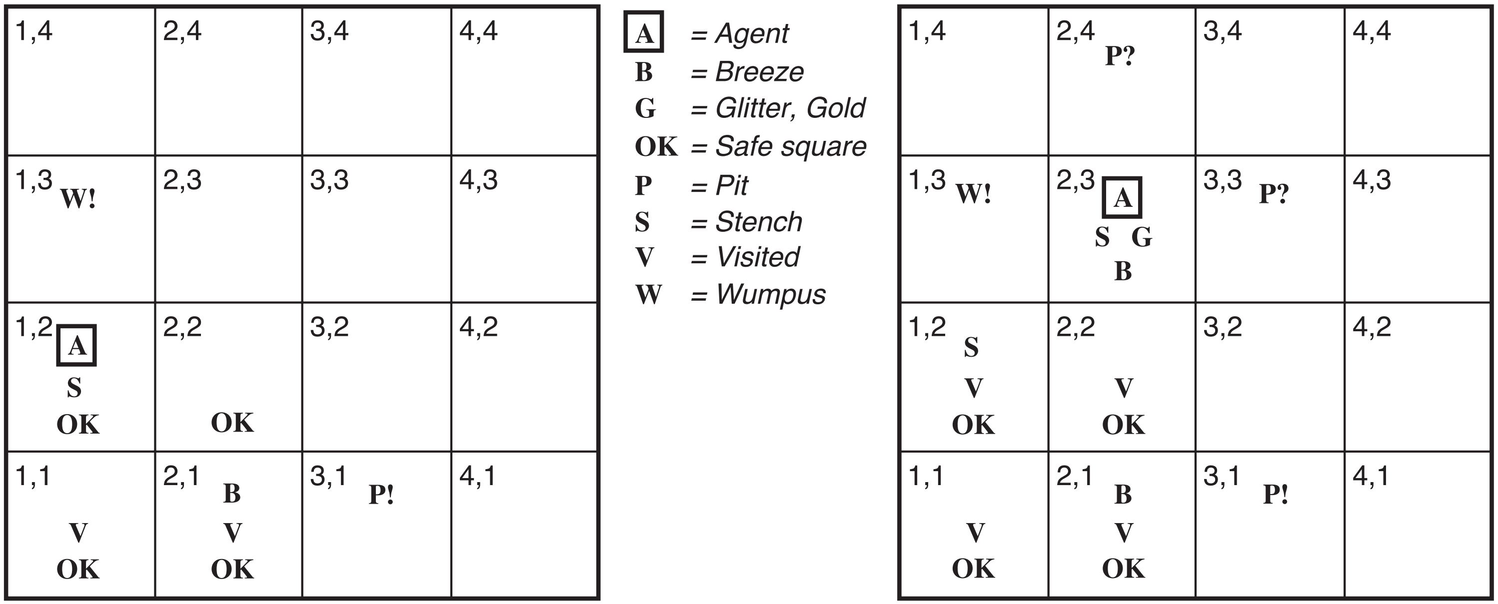
# Exploring in the Wumpus World




(a) (b)

- Suppose agent moves forward into [2,1]
- Perceives breeze, so there must be a pit in [3,1] or [2,2] (cannot be in [1,1])
- At this point only one known safe square not yet visited – [1,2]
- So agent backtracks and visits [1,2]

# Exploring in the Wumpus World




(a) (b)

- In [1,2] agent perceives [Stench,None,None,None,None]. What can it infer?
  1. There is a wumpus in [1,3] (there was no stench perceived in [2,1])
  2. There is a pit in [3,1] – difficult inference – based on previous inference that there is a pit in [2,2] or [3,1] and current inference that there is no pit in [2,2]
- Agent can now move to [2,2]. If it then choses to move to [2,3] it will find the gold

# Logic

- Note that in the example above, when the agent draws a conclusion from available information the conclusion is guaranteed to be correct
  - This is a fundamental property of logical reasoning
- Assume you know fundamentals of propositional and predicate logic from COM1002
- Will
  - Give brief overview of fundamental concepts of formal logic
  - Discuss use of formal logic for knowledge representation in AI
  - Describe how to construct a wumpus world agent that uses propositional logic

# Syntax and Semantics

- KB consists of sentences
- Sentences in formal logical languages are expressed according to the syntax of the language
  - Syntax defines what is a well-formed ("allowable") sentence in the representation language
  - E.g.
    - "x+y=4" is a well-formed sentence of arithmetic
    - "x4y+=" is not is a well-formed sentence of arithmetic
- A logic must also define the semantics or meaning of sentences
  - Semantics defines the truth of each sentence wrt a possible world
  - E.g. semantics of arithmetic specifies "x + y = 4" is
    - true in a world where x is 2 and y is 2
    - false in a world where x = 1 and y = 1
  - In standard logics every sentence must be either true or false in every possible world

# Models and Entailment

- The term "model" is used instead of "possible world" in formal logic
  - "possible world" suggests a potential real environment an agent might find itself in
  - "model" is a more precise, mathematical abstraction of this idea – a model fixes/assigns the truth or falsity of each sentence in the KB
- If sentence $\alpha$ is true in a model $m$ we say m satisfies $\alpha$ or $m$ is a model of $\alpha$
- $M(\alpha)$ refers to the set of all models of $\alpha$

# Models and Entailment

- Given a notion of truth can now talk about reasoning
- A sentence β logically follows from another sentence $\alpha$ or equivalently

  $\alpha$ entails β, written $\alpha \vDash \beta$,

  if and only if in every model in which $\alpha$ is true β is also true

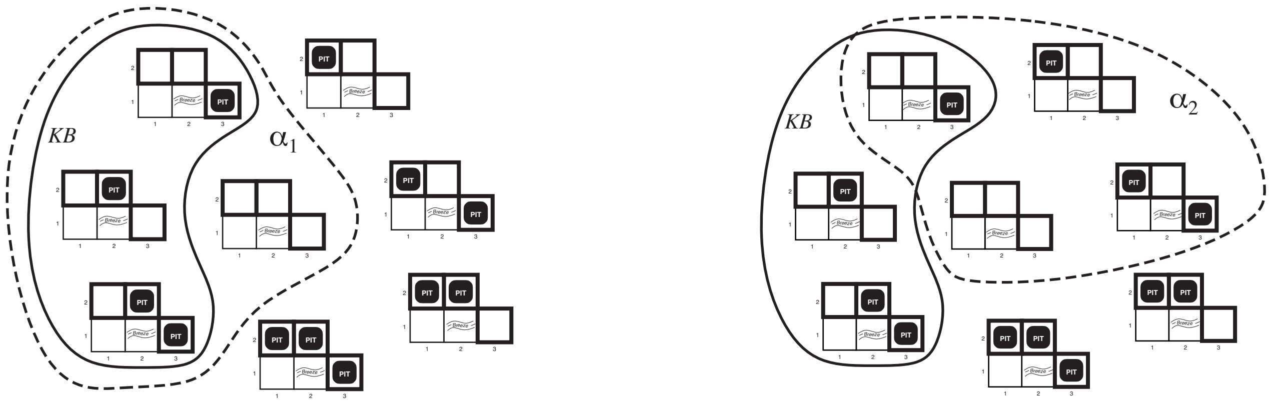
  – Informally: "whenever $\alpha$ is true, β is true too", or, "whenever $\alpha$ is true it is impossible for β to be false"
- More formally this is written as:

  $\alpha \vDash \beta$ if and only if $M(\alpha) \subseteq M(\beta)$
- Note if $\alpha \vDash \beta$ then $\alpha$ is a stronger assertion than β – it rules out more possible worlds

# Entailment in the Wumpus World

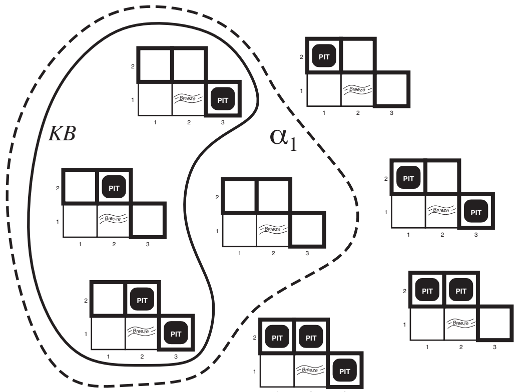
- Consider the wumpus world example above where the agent detected nothing in [1,1] and a breeze in [2,1]
- KB at this point consists of these two percepts + the rules of the wumpus world
- Agent wants to know if [1,2], [2,2] or [3,1] contain pits
- Each might or might not contain a pit
  - So there are $2^3 = 8$ possible models

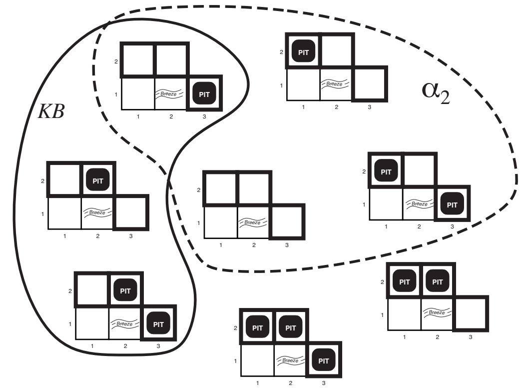
# Entailment in the Wumpus World




- KB can be thought of as a set of sentences
- KB is false, i.e. contains false sentences, in models that contradict what agent knows
  - E.g. KB is false in any model where [1,2] contains a pit, since no breeze was perceived in [1,1]
- Only 3 (of 8) models where KB is true
  - Surrounded by solid line

# Entailment in the Wumpus World

(a) 


(b) 


- Consider

  $\alpha_1$ = "There is no pit in [1,2]"

  $\alpha_2$ = "There is no pit in [2,2]"
- Models of $\alpha_1$ are those in (a) surrounded by dotted lines; models of $\alpha_2$ are similarly shown in (b)
- By visual inspection can see that in every model in which KB is true, $\alpha_1$ is also true
  - so KB $\models \alpha_1$ and agent can conclude there is no pit in [1,2]
- Can also see that in some models in which KB is true $\alpha_2$ is false
  - so KB $\not\models \alpha_2$ and agent cannot conclude there is no pit in [2,2] (or that there is)