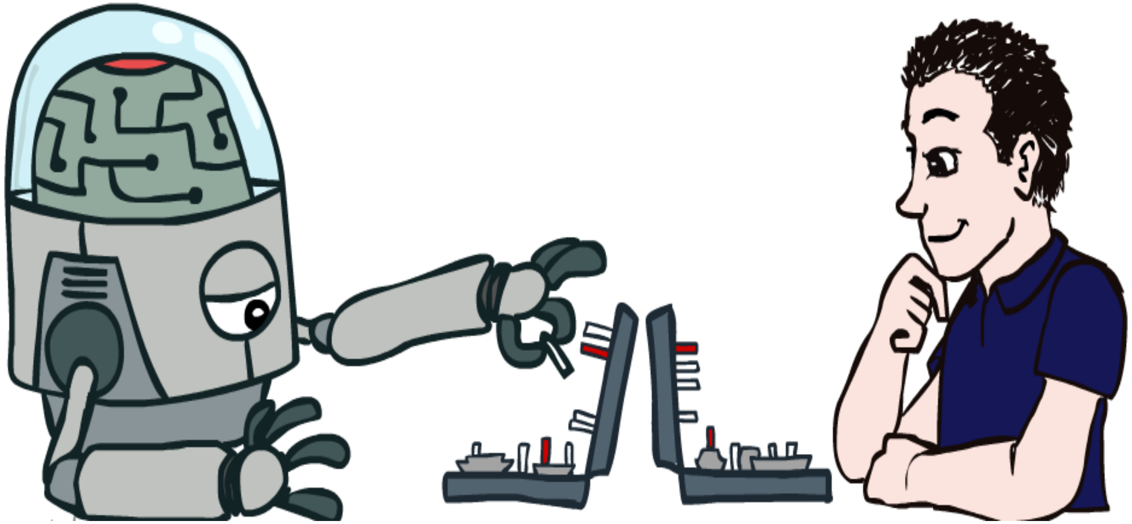
ARTIFICIAL INTELLIGENCE- CS411

Dr. Abdulelah Algosaibi



Credit: Dr. Alaa Sagheer

Logic & Inference

Use inference to derive new representations, then deduce what to do

Lecture I

Introduction

- Reflex agents find their way from Arad to Bucharest by dumb luck,
- Drawing reasonable conclusions from a set of data (observations, beliefs, etc.) seems key to intelligence
- Logic is a powerful and well developed approach to this and highly regarded by people
- Logic is also a strong formal system that we can program computers to use
- Maybe we can reduce any AI problem to figuring out how to represent it in logic and apply standard proof techniques to generate solutions

Introduction

❖ **Knowledge and Reasoning:**

- We will discuss here- *the representation of knowledge and the reasoning processes* that bring knowledge to life. (Central of AI).
- Humans know things and do reasoning. Knowledge and reasoning are also important for artificial agents because they enable successful behaviors that would be very hard to achieve otherwise.
- We have seen that knowledge of action outcomes enables problem-solving agents to perform well in a fully observable environment (Arad-Bucharest)
- Knowledge and reasoning also play a crucial role in dealing with partially observable environments.
- A knowledge-based agent combines general knowledge with current percepts to **infer** hidden aspects of the current state prior to selecting actions. **(e.g. Physician)**

Introduction

❖ **Knowledge and Reasoning:**

- Another reason for studying knowledge-based agents is their **flexibility**. They are able to accept new tasks in the form of explicitly described goals,
  - Also, they can achieve competence quickly by being told or learning new knowledge about the environment, and they can adapt to changes in the environment by updating the relevant knowledge.

Introduction *cont.*

- **Knowledge and Reasoning:**
  - How can we formalize our knowledge about the world so that:
    - We can reason about it
    - We can do sound inference
    - We can prove things
    - We can plan actions
    - We can understand and explain things
  - In this chapter:
    - **7-1 Definition of Knowledge-based Agents**
    - **7.2 The Wumpus World**
    - **7-3 Logic**
    - **7-4 Propositional Logic**
    - **7-5 First order Logic**

Introduction

❖ **Knowledge-based Agent** ← Deep Learning

- Logical (Knowledge-Based) agents combine general knowledge with current percepts to infer hidden aspects of current state prior to selecting actions.
- Agent that uses prior OR/AND acquired knowledge to achieve its goals and make more efficient decisions
- Knowledge-based agents are best understood as agents that know about their world and reason about their courses of action.

They have a aue.

Basic concepts:

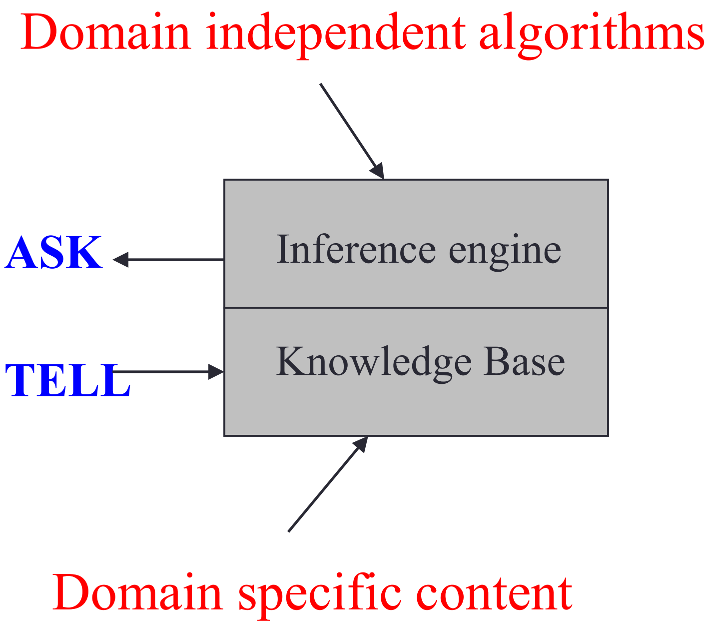
- The knowledge-base (KB): a set of representations of facts about the world.
- The knowledge representation language: a language whose sentences represent facts about the world.

KB → Knowledge Base => Information Given to Agent

Knowledge-based Agents

❖ **The central component of a knowledge-based agent is its knowledge base or KB.**

- A KB is a set of representations (sentences) of facts about the Agent environment.
- Use some knowledge representation language, to **TELL** it what the agent knows, i.e. to add new knowledge to the KB
- ASK agent to query what to do (or to query what is known to the KB)
- Agent can use inference to deduce new facts from the TELLed facts



Generic KB Agent

```
function KB-AGENT(percept) returns an action
    static: 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 ← t + 1
    return action
```

❖ Each time the agent program is called, it does three things.

KB is essentially an monk ← Nb.

1. TELL KB what was perceived

   Insert new sentences, representations of facts, into KB

   <u>MAKE-PERCEPT-SENTENCE</u> constructs a sentence asserting that the agent perceived the given percept at the given time.

2. ASK KB what to do.

   Use logical reasoning to examine actions and select best.

   <u>MAKE-ACTION-QUERY</u> constructs a sentence that asks what action should be done at the current time.

Generic KB Agent

function KB-AGENT(*percept*) **returns** an *action*
  static: *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*

❖ Each time the agent program is called, it does three things.

3. Record its choice with TELL and executes the action.

KB knows that the hypothetical action has actually been executed.

MAKE-ACTION-SENTENCE constructs a sentence asserting that the chosen action was executed

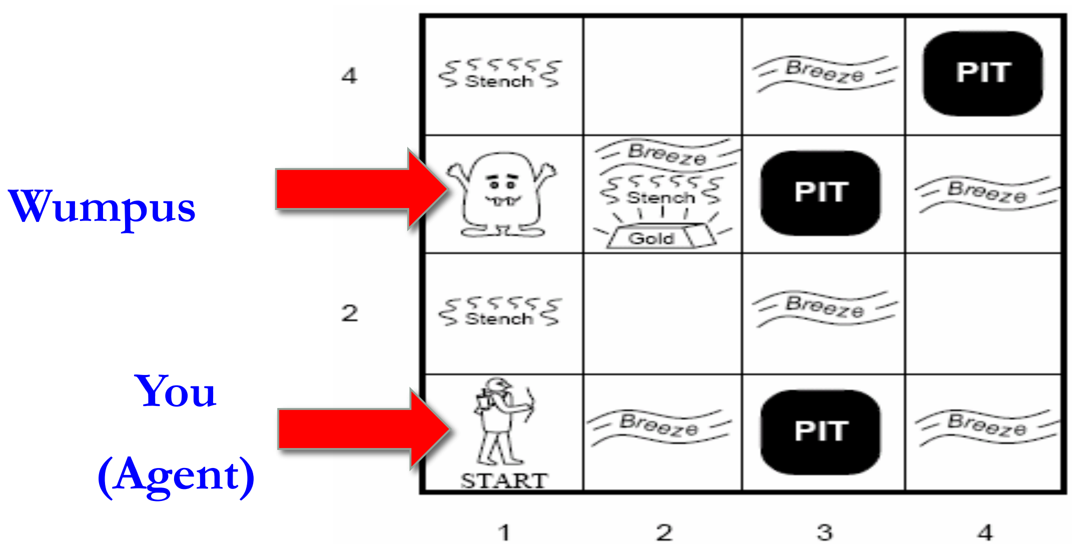
The Wumpus World

A toy world for logical exploration

Computational Systems that
"Think" Rationally

The Wumpus World

- A four by four cave with locations identified by coordinates (3,4), etc.
- Agent is at a location, facing a particular direction (L,R,D,U)
  - Agent starts at (1,1) facing R



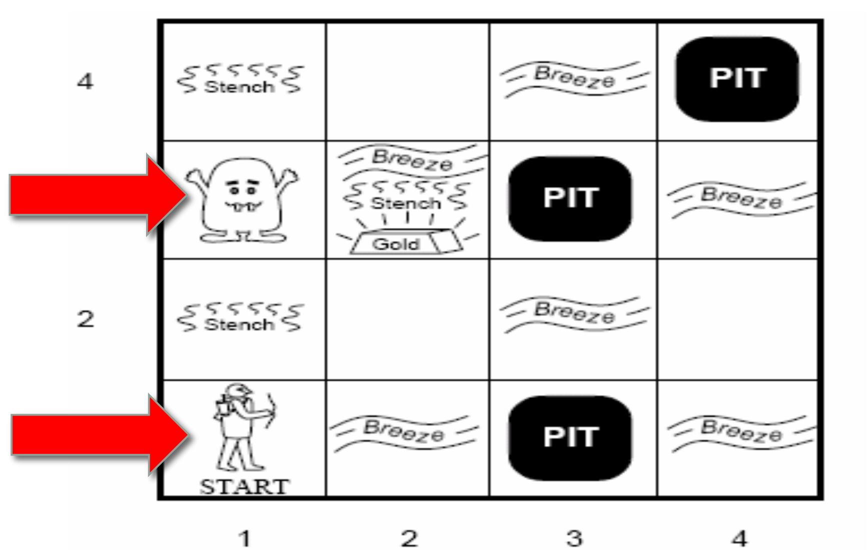
The Wumpus World

- In the cave is:
  - A **Wumpus** that smells:
    - It can kill the agent if at same location,
    - It can be killed by the agent shooting an arrow if facing the Wumpus. When the Wumpus dies, it SCREAMs! ← ?

Wumpus

You

(Agent)



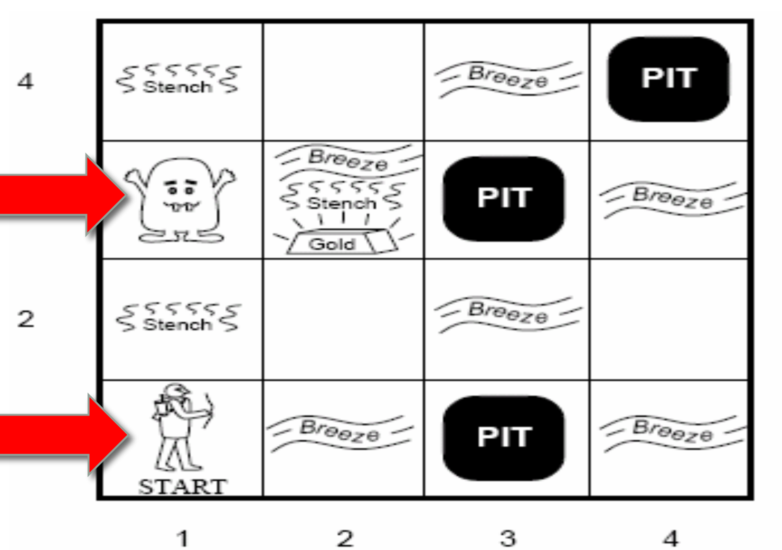
The Wumpus World

- In the cave are:
  - 3 Pits. Breezes blow from pits.
    - If an agent steps into a pit, it falls to its death.
  - A heap of gold that glitters

Wumpus

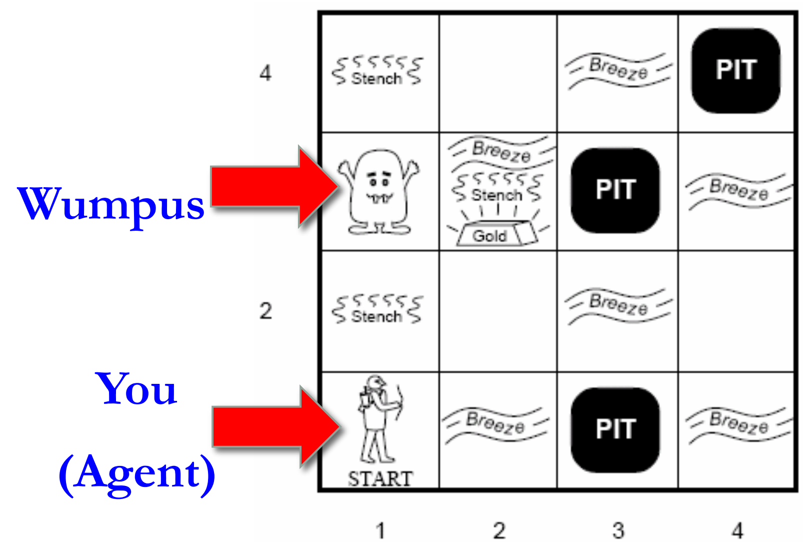
You

(Agent)



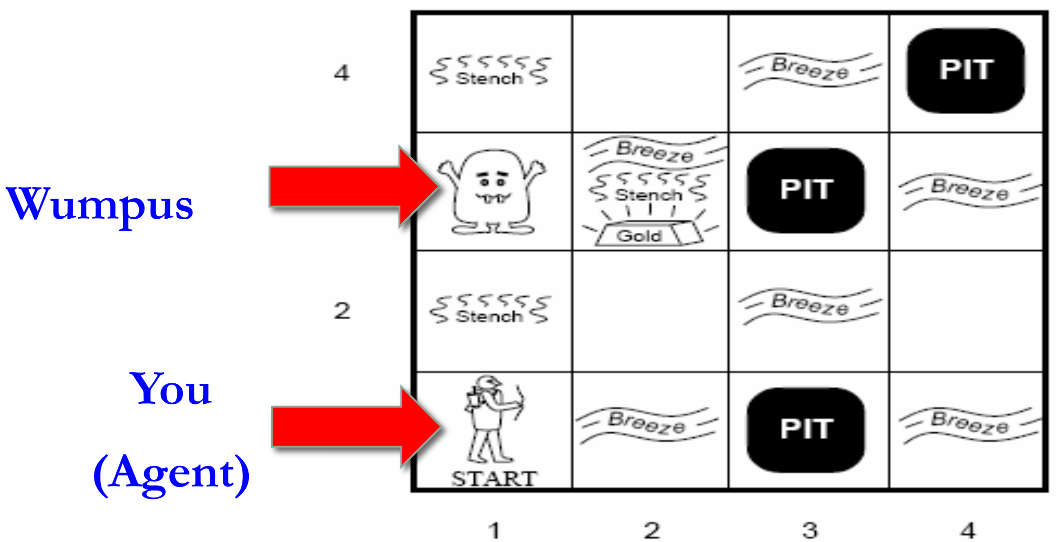
The Wumpus World

- Agent goal:
  - get gold and get out alive
- Agent actions:
  - Move forward one square in current direction (Fwd)
  - Turn left or right 90^o (TL,TR)
  - Shoot arrow in current direction
  - Grab gold
- Agent perceptions at each location:
  - [Stench, Breeze, Glitter, Bump, Scream]



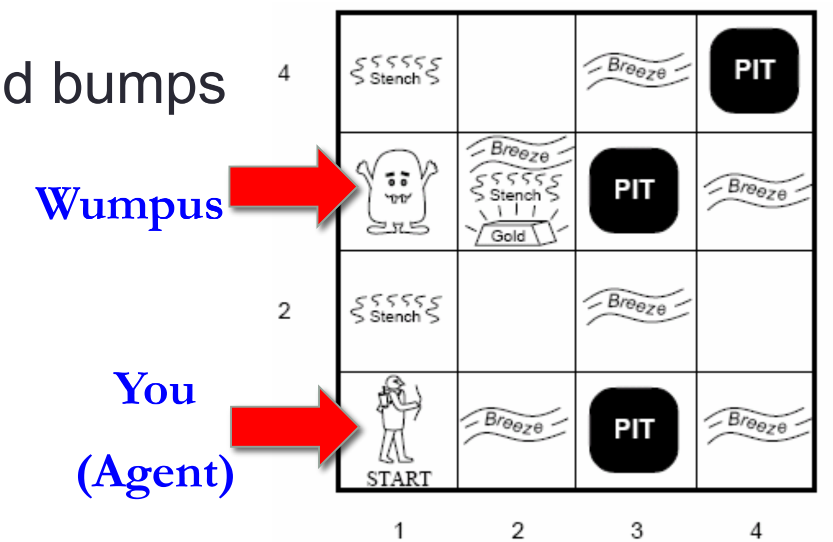
The Wumpus World *

- ❖ Cave is created randomly (location of Wumpus, pits and gold)
- ❖ Agent must construct a model – knowledge base about the cave – as it tries to achieve its goal



The Wumpus World

- General knowledge (known at start)
  - Location and direction
  - Living
  - Grab and holding
  - Wumpus and stench, shooting, scream, life
  - Pits and breeze
  - Gold and glitter
  - Movement and location, direction and bumps
  - Starting state of agent
  - Goal
- Facts (not known)
  - Location of Wumpus, pits, gold

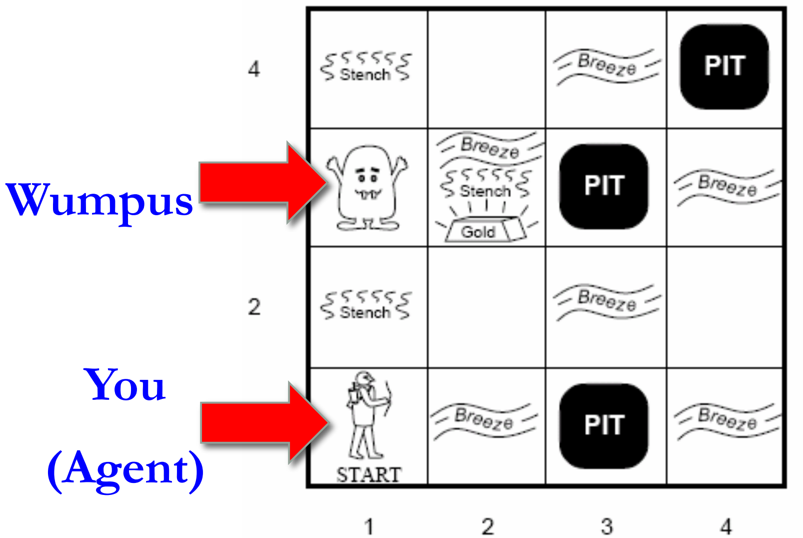


PEAS

Performance measure

gold +1000, death -1000

-1 per step, -10 for using the arrow



PEAS

Performance measure

gold +1000, death -1000

-1 per step, -10 for using the arrow

Environment

Squares adjacent to wumpus are smelly

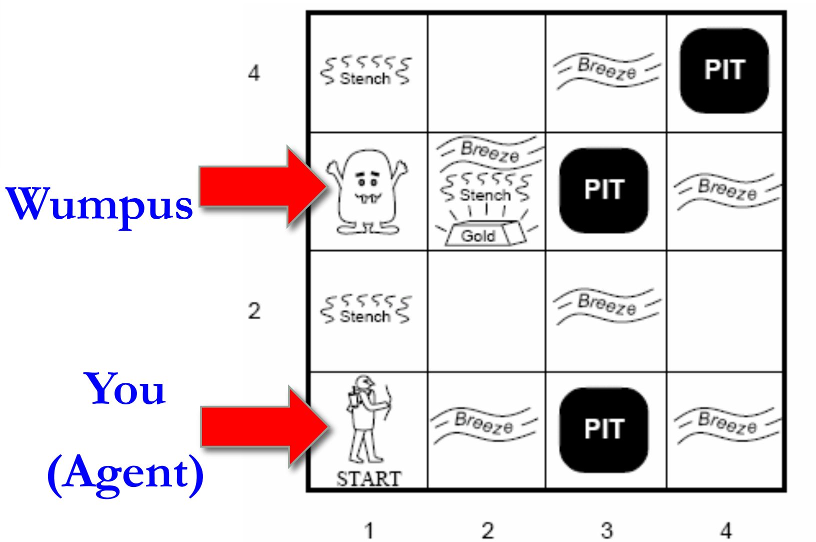
Squares adjacent to pit are breezy

Glitter iff gold is in the same square

Shooting kills wumpus if you are facing it

Shooting uses up the only arrow

Grabbing picks up gold if in same square



PEAS

Least steps and least Arrow usage And No Deaths.

Performance measure

gold +1000, death -1000

-1 per step, -10 for using the arrow

Environment

Squares adjacent to wumpus are smelly

Squares adjacent to pit are breezy

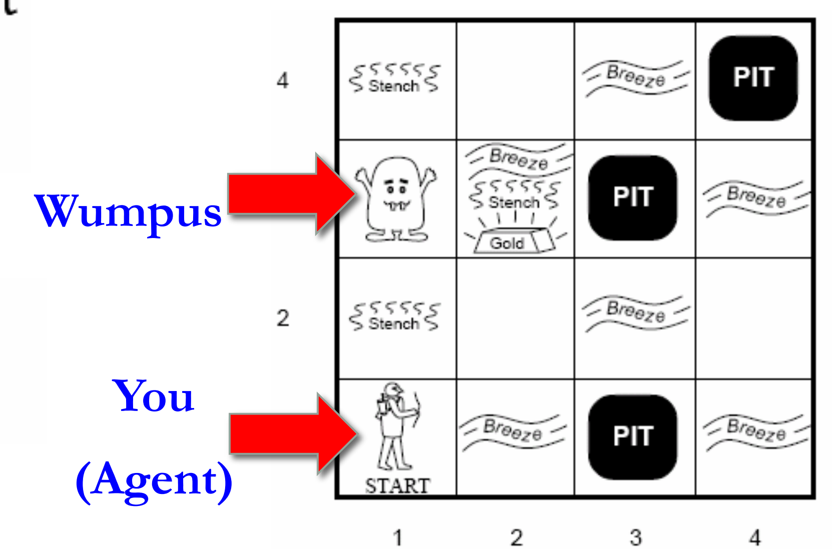
Glitter iff gold is in the same square

Shooting kills wumpus if you are facing it

Shooting uses up the only arrow

Grabbing picks up gold if in same square

Actuators Left turn, Right turn,
Forward, Grab, Release, Shoot



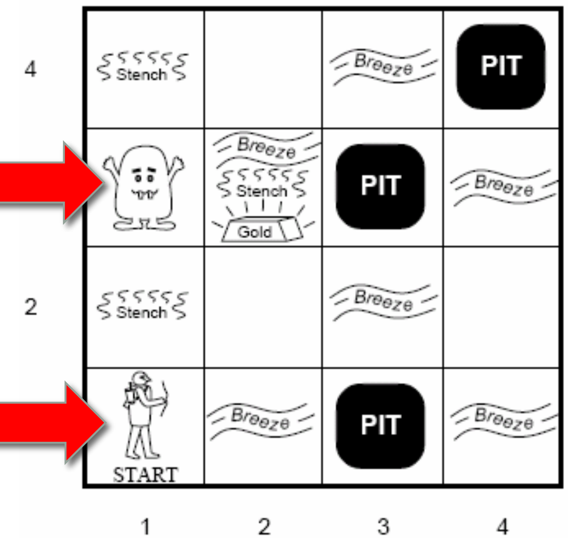
PEAS

Goal: bring back gold as quickly as possible

Performance measure
- gold +1000, death -1000
- -1 per step, -10 for using the arrow

Environment
- Squares adjacent to wumpus are smelly
- Squares adjacent to pit are breezy
- Glitter iff gold is in the same square
- Shooting kills wumpus if you are facing it
- Shooting uses up the only arrow
- Grabbing picks up gold if in same square

Wumpus



You (Agent)

Actuators Left turn, Right turn, Forward, Grab, Release, Shoot

Sensors: Stench, Breeze, Glitter, Bump, Scream

Acting & Reasoning

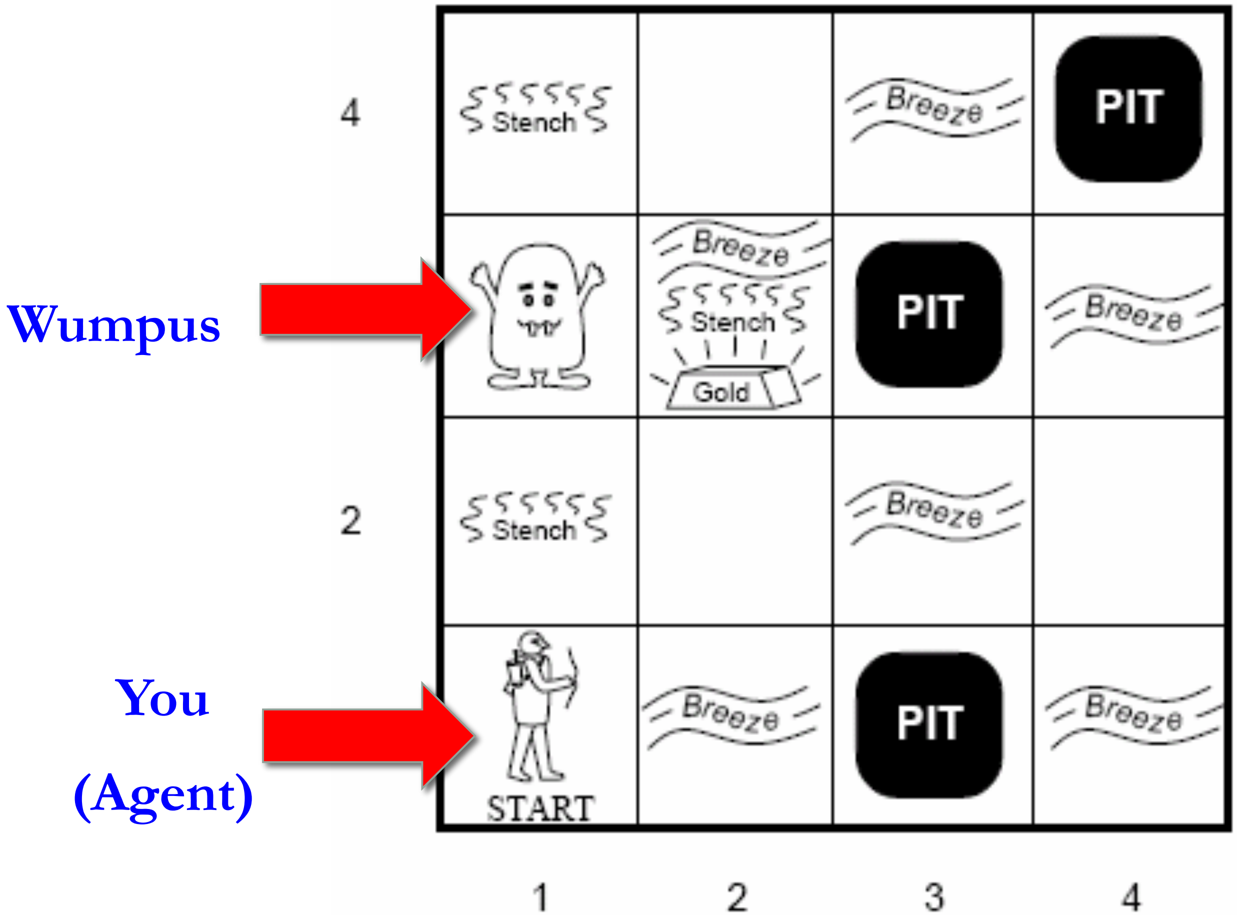
Dicasing Thru Breeze + Smell

- Fully Observable? **No – only local perception**
- Deterministic? **Yes – state+action determines outcome specifically**
- Static? **Yes – Wumpus and pits do not move**
- Discrete? **Yes, no real-valued states or actions**
- Episodic? **No – sequential at the level of actions (involves a sequence of actions)**
- Single Agent **Yes – Wumpus is essentially a natural feature**

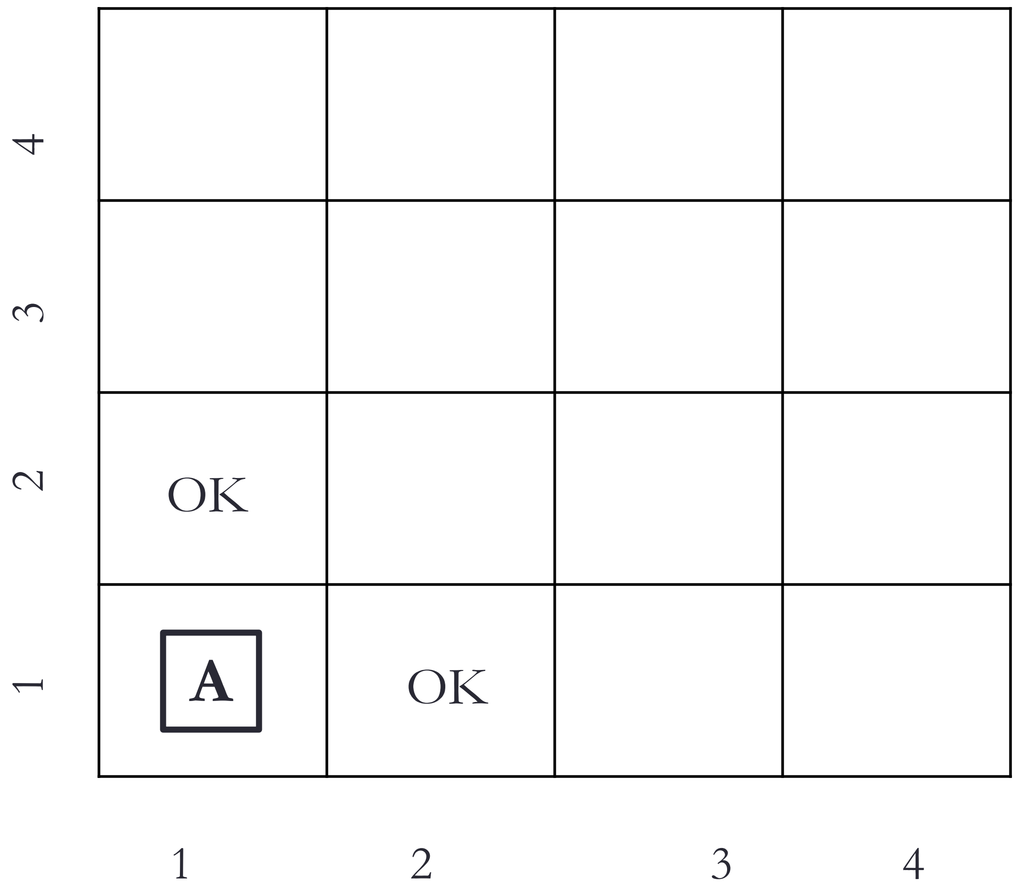
Acting & Reasoning *

- Let' s play the Wumpus game!
- The conclusion: "what a fun game!"
  - Another conclusion: If the available information is correct, the conclusion is guaranteed to be correct.

Play with Wupmus!

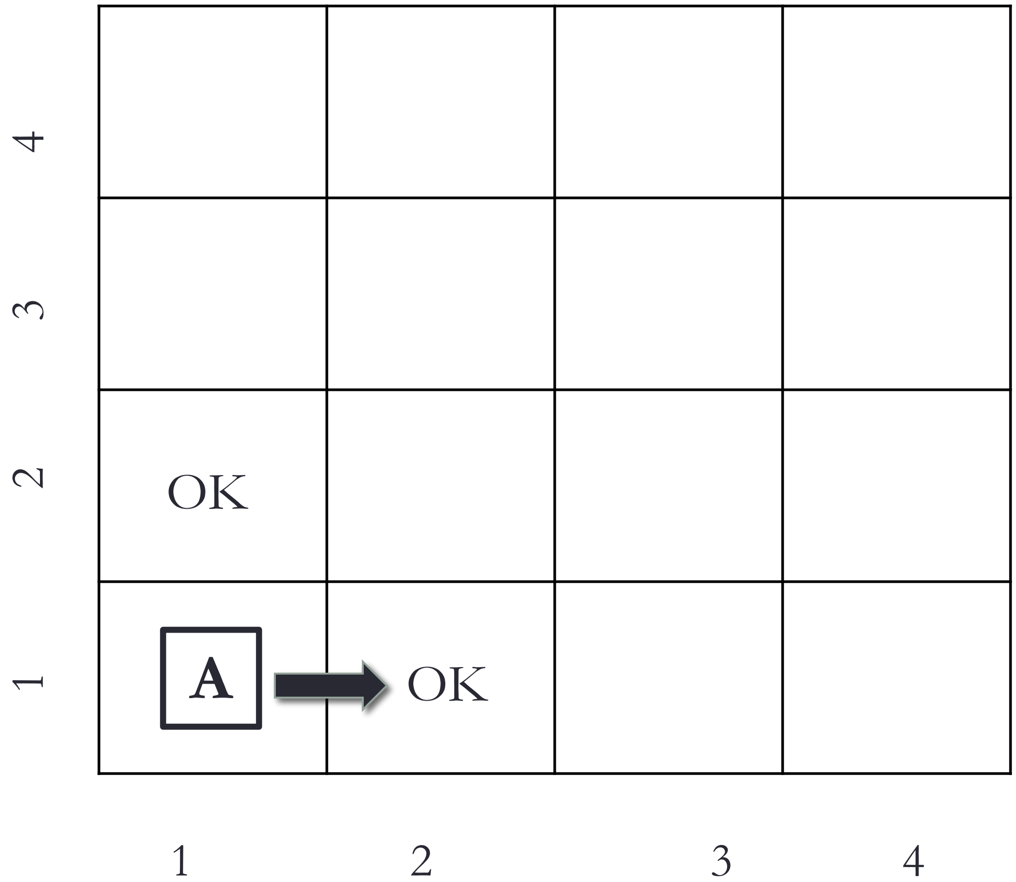


Play with Wupmus!



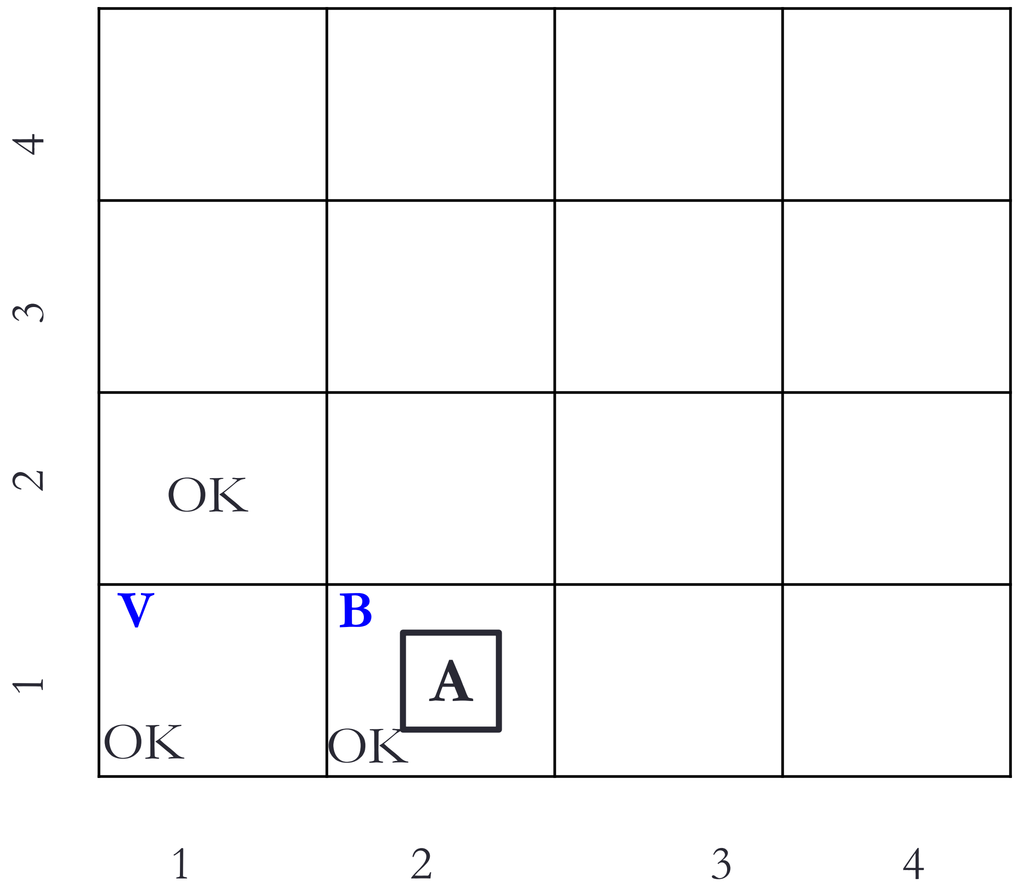
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



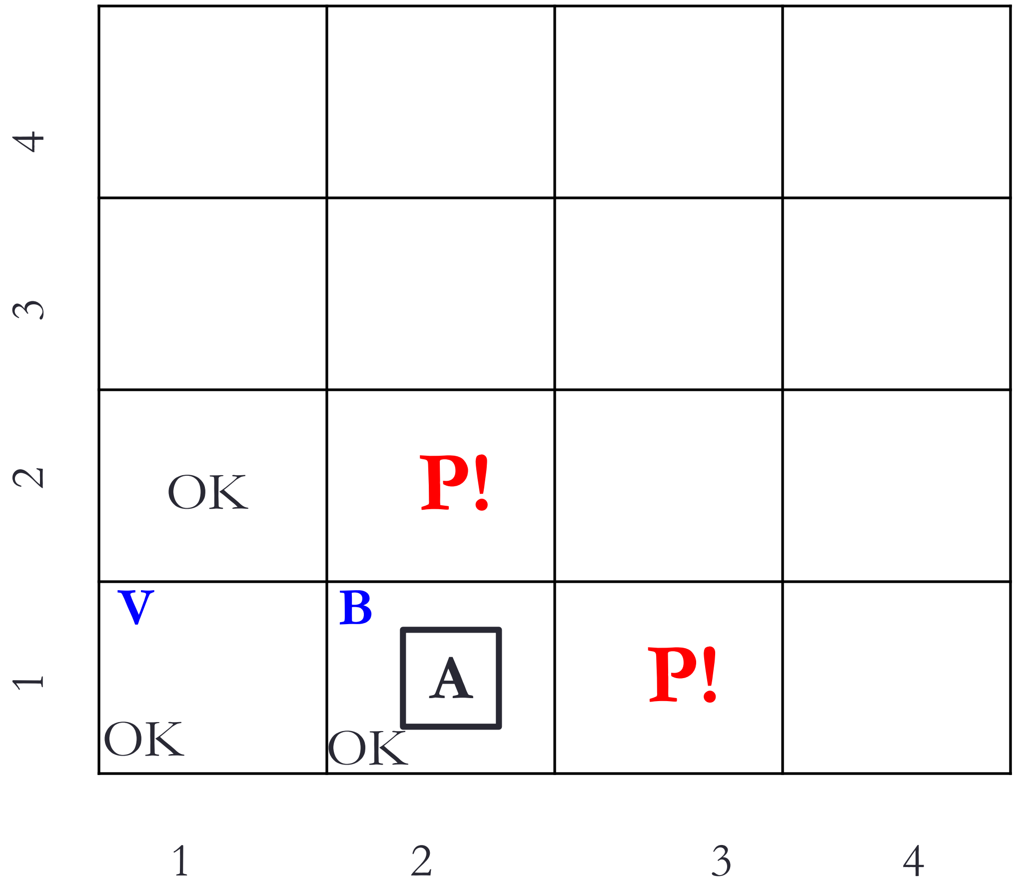
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



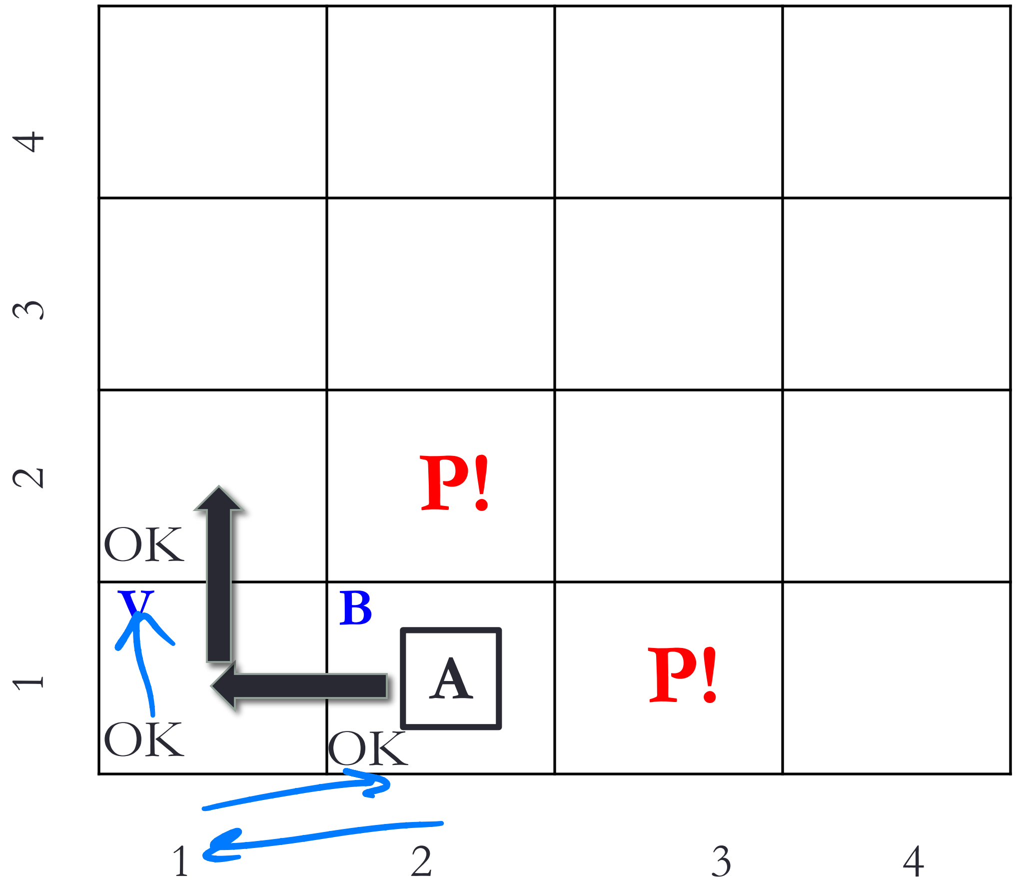
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



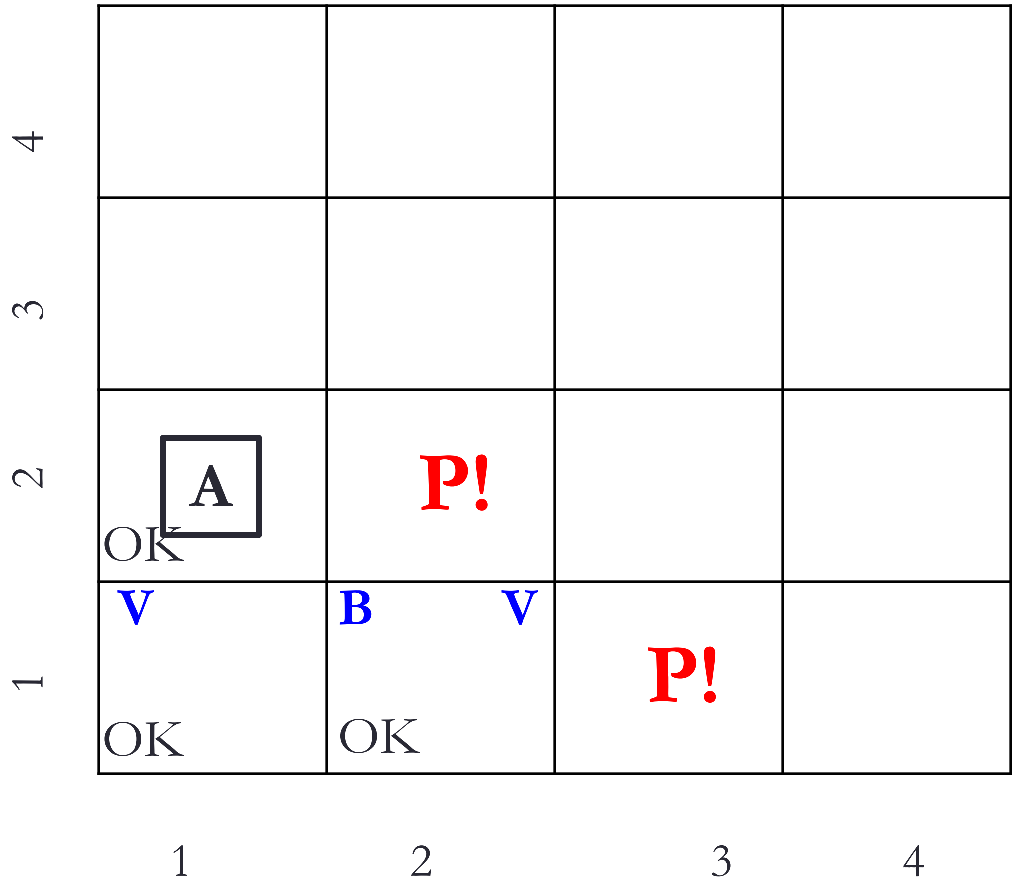
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



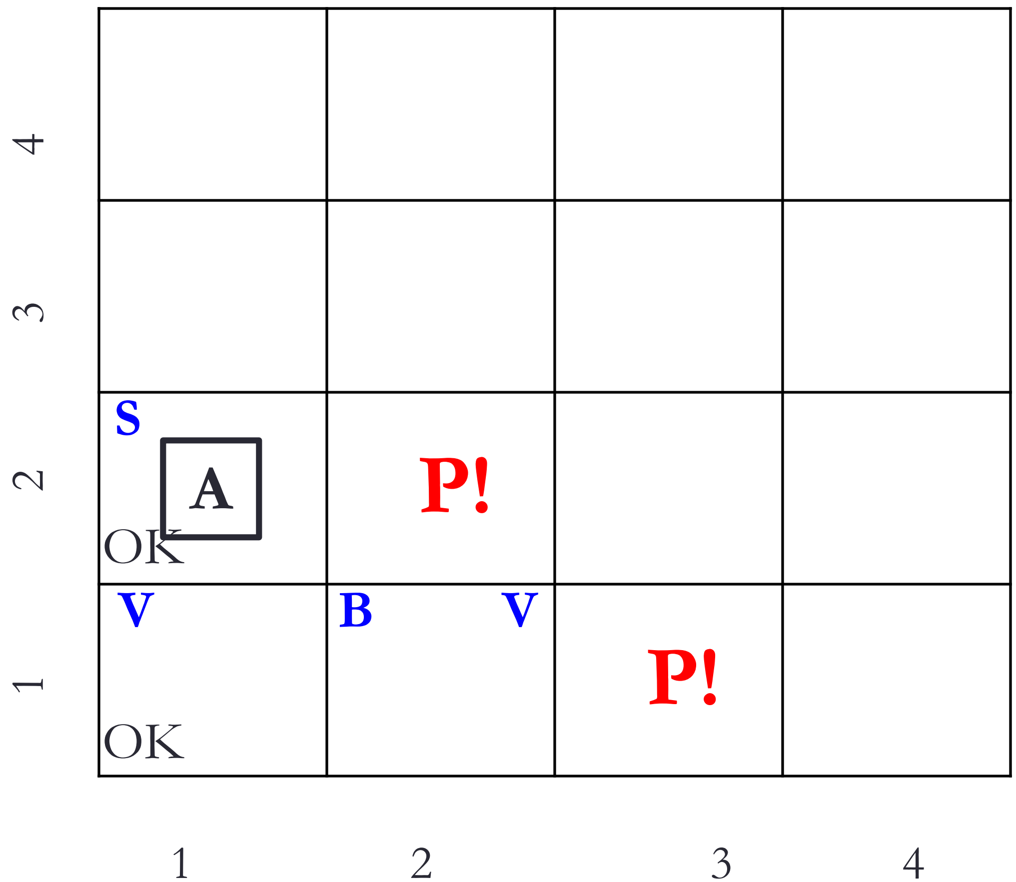
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



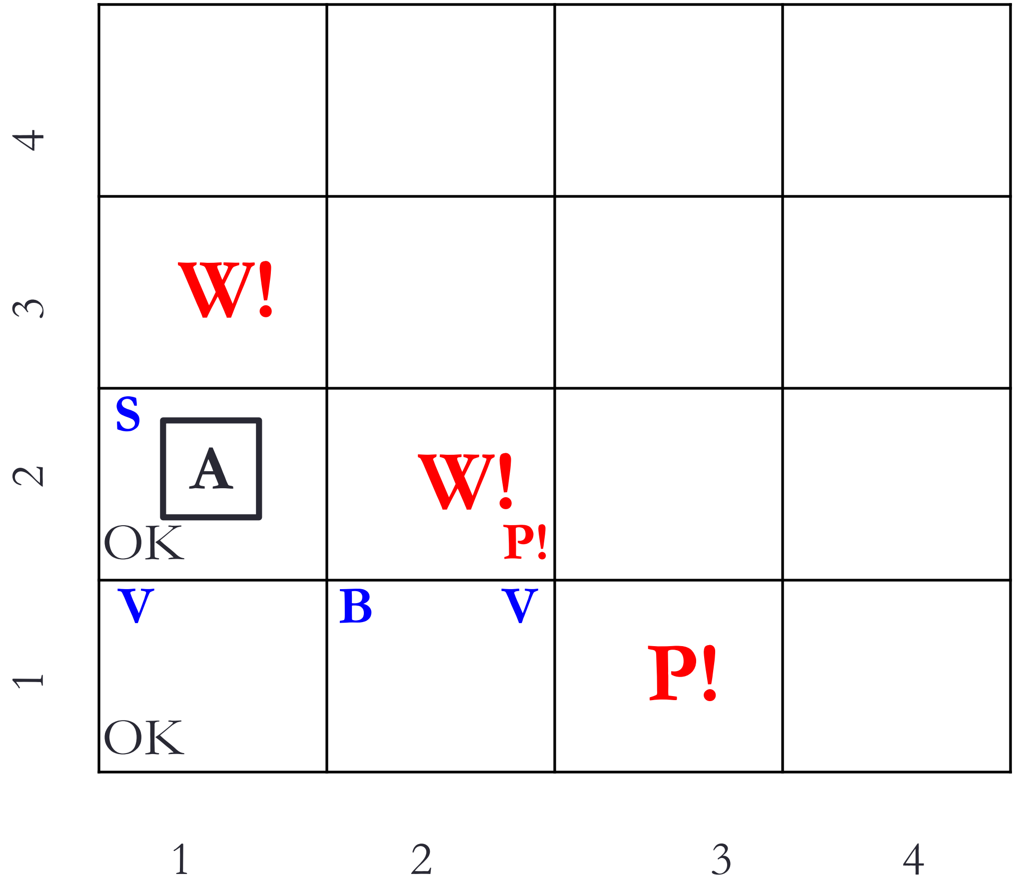
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



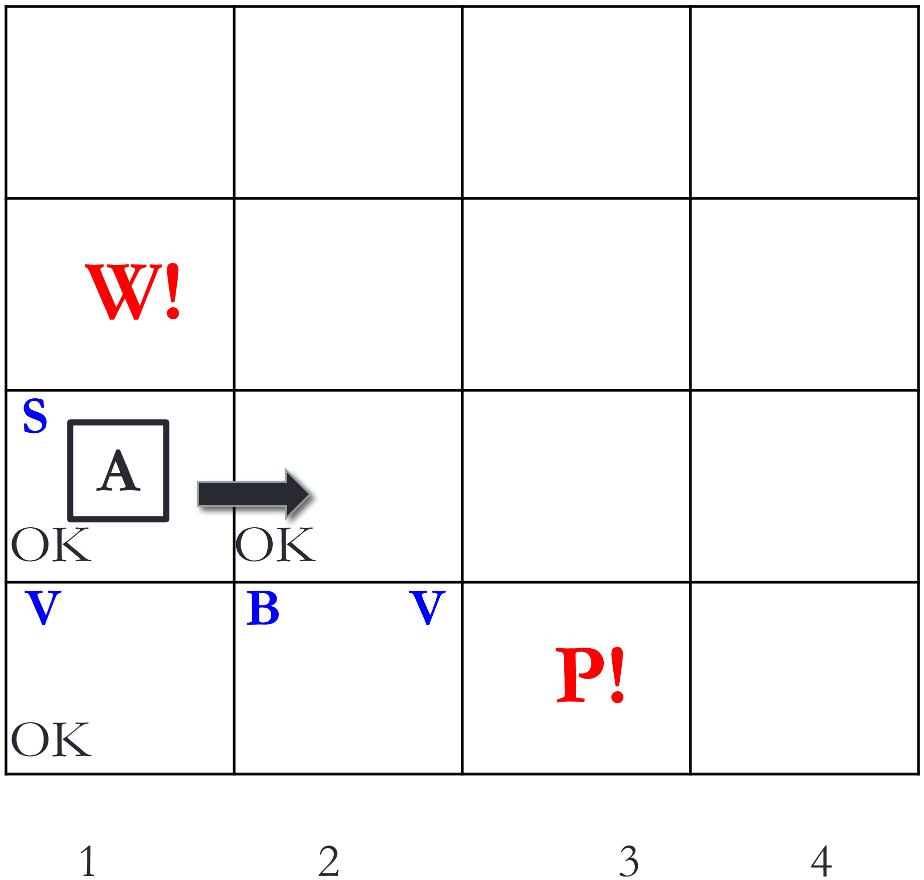
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



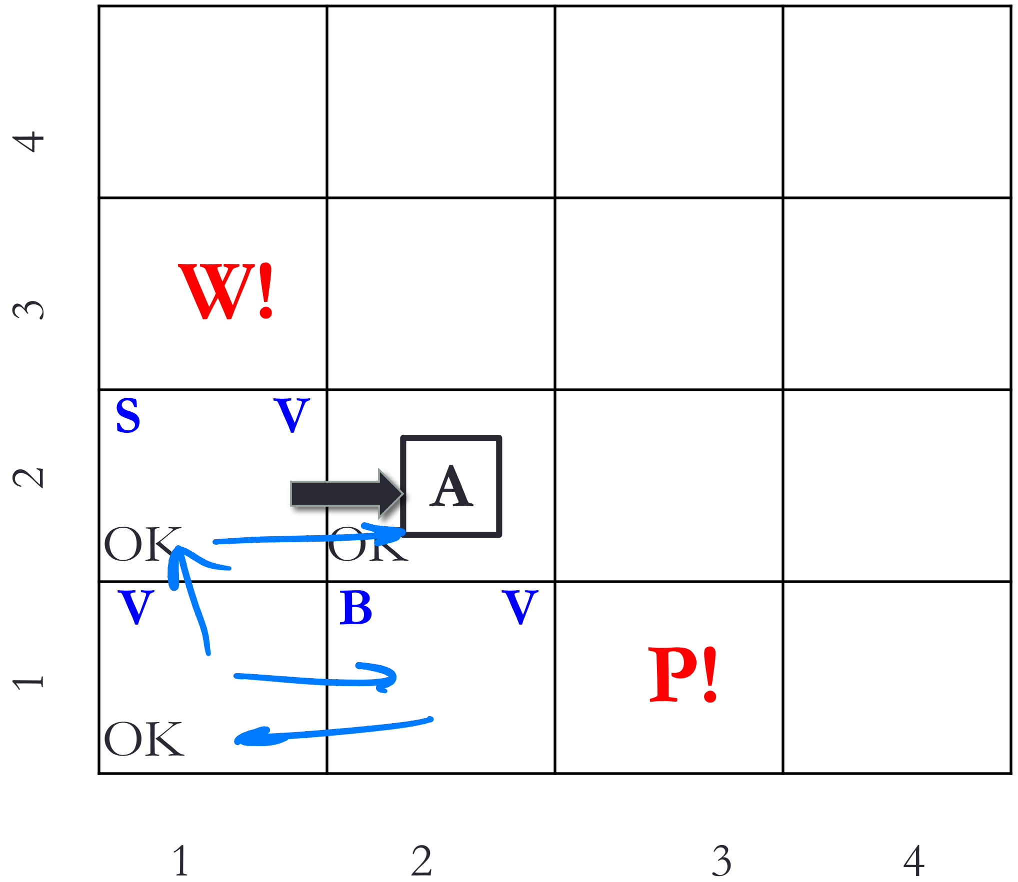
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



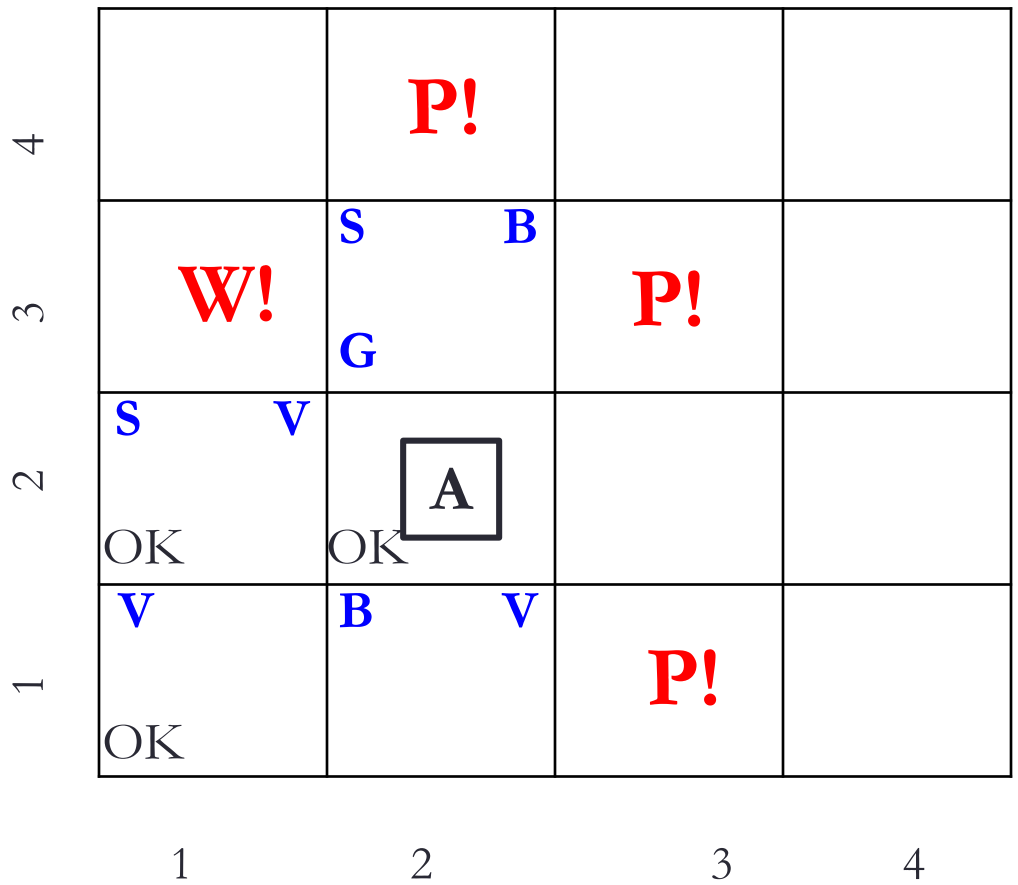
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



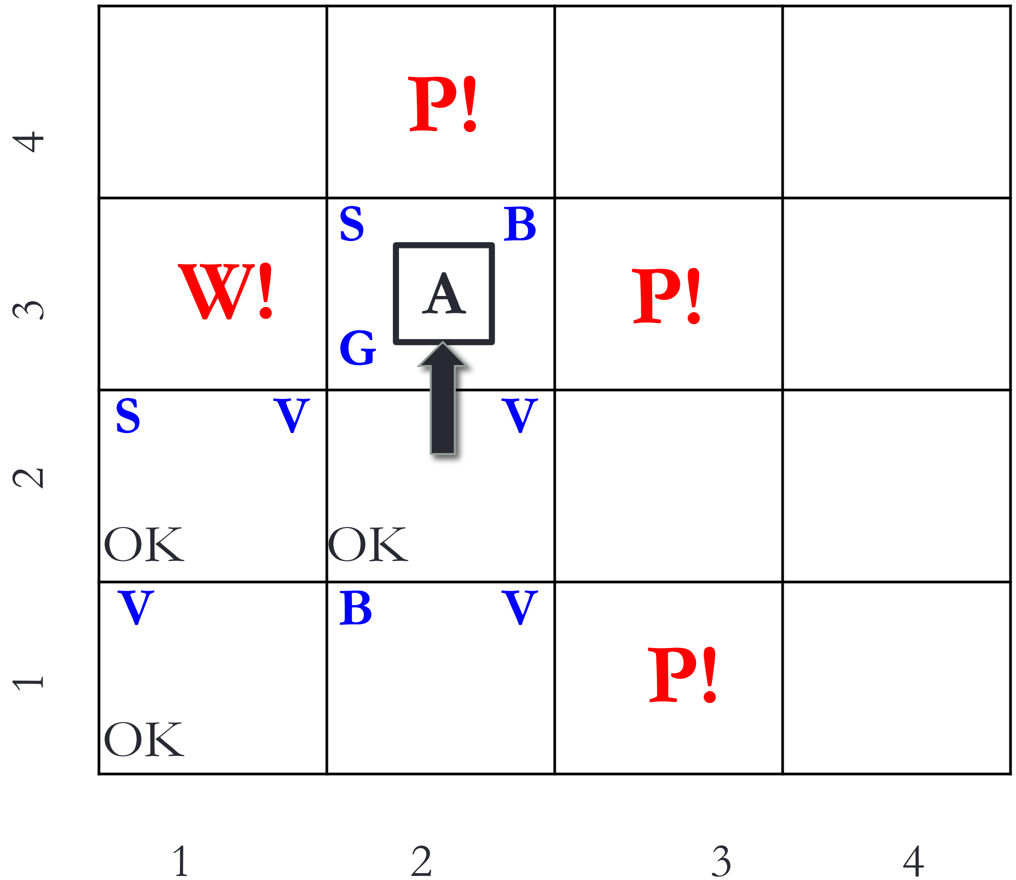
A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Play with Wupmus!



A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

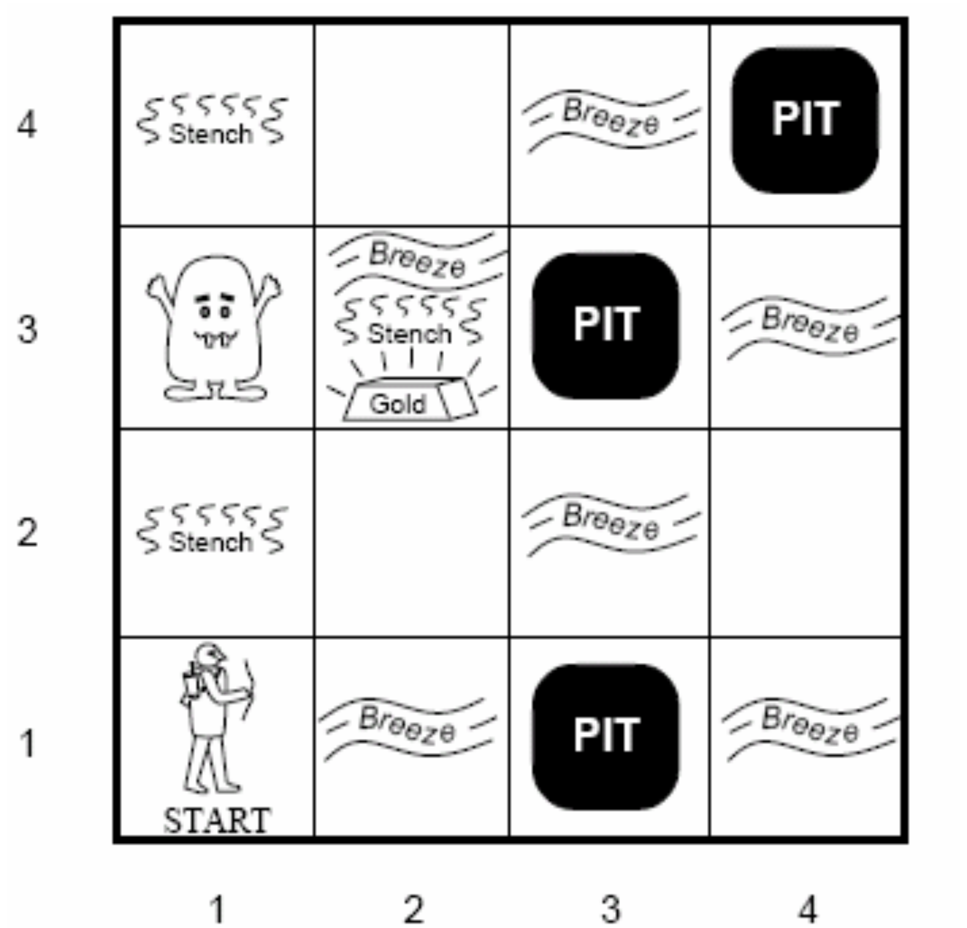
Play with Wupmus!



A= Agent
B= Breeze
S= Smell
P= Pit
W= Wumpus
OK = Safe
V = Visited
G = Glitter

Logical Reasoning

In each case where the agent draws a conclusion from the available information, that conclusion is guaranteed to be correct if the available information is correct. This is a fundamental property of logical reasoning.



Knowledge-based Agent and Rationality

A rational agent is an agent that has clear preferences and always chooses to perform the action with the optimal expected outcome for itself from among all feasible actions.

A knowledge-based agent includes a knowledge base and an inference engine. The inference engine applied logical rules to the knowledge base and deduce new knowledge.

Then, it is ….?

6. Logic and Inference

Foundations of Logic

Lecture II

LOGIC!

Logic → Formal Language for Information Representation.

Logics are formal languages for representing information such that conclusions can be drawn

Syntax defines the sentences in the language (defines structure of sentences)

Semantics define the "meaning" of sentences; i.e., define truth of a sentence in a world

Semantical Definition of a Sentences.

E.g., the language of arithmetic

$x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence

$x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y

$x + 2 \geq y$ is true in a world where $x = 7,\ y = 1$
$x + 2 \geq y$ is false in a world where $x = 0,\ y = 6$

Models

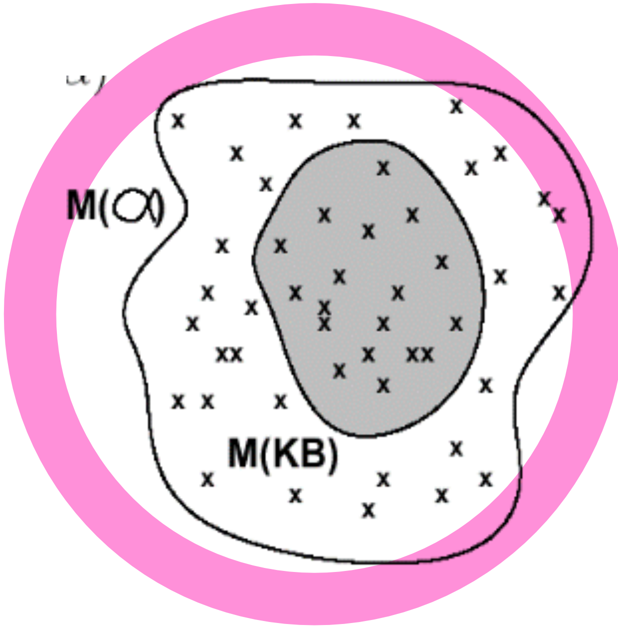
- A **model** is a possible state of the world

  Model

- If a sentence α is true in model *m*, then *m* is **a model** of α

  Or…. m is a model of α if α is true in m.

- *M*(α) means the set of **all models** of α.



Models

Model is a possible state of the world

- A **model** is a possible state of the world
- If a sentence α is true in model *m*, then *m* is **a model** of α
- *M*(α) means **all models** of α

Example: Arithmetic

Given the sentence α which looks like $x + 2 > y$:

Models *

- A **model** is a possible state of the world
- If a sentence α is true in model *m*, then *m* is **a model** of α
- *M*(α) means **all models** of α

Example: Arithmetic

Given the sentence α which looks like $x + 2 > y$:

- **Is {x=7, y=1} a model of α?**

Models *

- A **model** is a possible state of the world
- If a sentence α is true in model *m*, then *m* is **a model** of α
- *M*(α) means **all models** of α

Example: Arithmetic

Given the sentence α which looks like $x + 2 > y$:

- **Is $\{x=7, y=1\}$ a model of α?** Yes

Models *

- A **model** is a possible state of the world
- If a sentence α is true in model *m*, then *m* is **a model** of α
- *M*(α) means **all models** of α

Example: Arithmetic

Given the sentence α which looks like $x + 2 > y$:

- **Is {x =7, y =1} a model of α? Yes**
- **Is {x =3, y =6} a model of α?**

Models

Important

- A **model** is a possible state of the world
- If a sentence α is true in model *m*, then *m* is **a model** of α
- *M*(α) means **all models** of α

Example: Arithmetic

Given the sentence α which looks like *x* + 2 > *y* :

- **Is {*x* =7, *y* =1} a model of α? Yes**
- **Is {x =3, y =6} a model of α? No**

A world where α is true is M(α)

If it is true in the world of m then m is a model of A

Entailment

- It means that one thing follows from another:

$$KB \models \alpha$$

- Knowledge base *KB* entails sentence α if and only if α is true in all worlds where *KB* is true. (Example: s = There is no Pit in [2,1])
  - E.g., the KB containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"
  - E.g., x+y = 4 entails 4 = x+y
  - Entailment is a relationship between sentences that is based on semantics

Entailment

Idea:

A sentence β entails a sentence α if and only if:

α is true in all worlds where β is true

Entailment

Idea

A sentence β entails a sentence α if and only if:

α is true in all worlds where β is true

Definition

β |= α if and only if

$$M(\beta) \subseteq M(\alpha)$$

Entailment

Idea

A sentence β entails a sentence α if and only if:

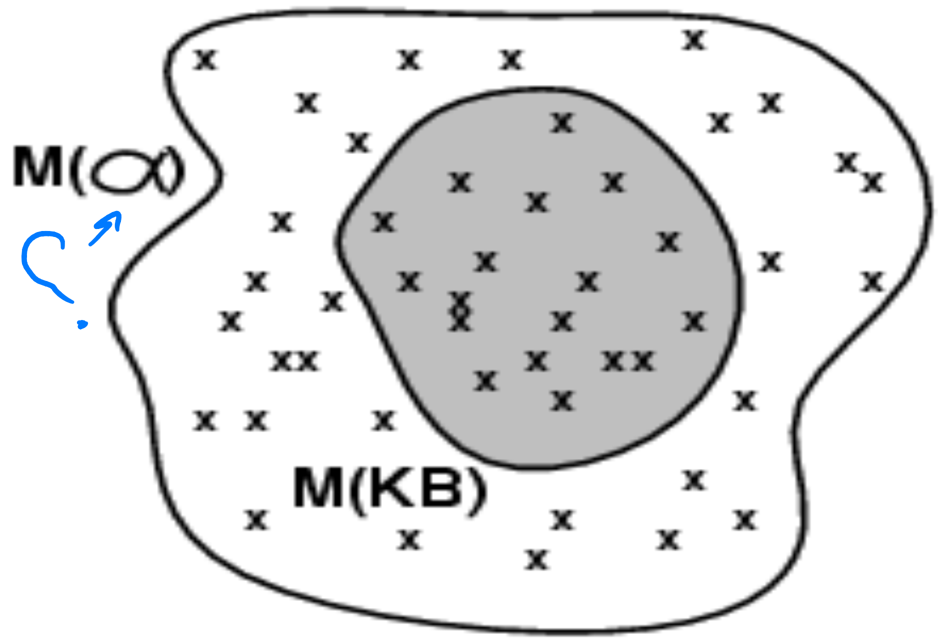
α is true in all worlds where β is true

Definition

β |= α if and only if

$$M(\beta) \subseteq M(\alpha)$$

M(KB) ⊆ M(α)



Entailment

Definition

β |= α if and only if $\boxed{M(\beta) \subseteq M(\alpha)}$

$\beta \models \alpha$ is $m(\beta) \subseteq m(\alpha)$

Entailment

Definition

$\beta \models \alpha$ if and only if $M(\beta) \subseteq M(\alpha)$

Example: Arithmetic

$(x + y = 4) \models (4 = x + y)$

$\beta \models A$ if & only if $M(\beta) \subseteq M(\alpha)$

Entailment

Definition

$\beta \models \alpha$ if and only if $M(\beta) \subseteq M(\alpha)$

Example: Arithmetic

$(x + y = 4) \models (4 = x + y)$

Example: English

"The **Broncos** won **and** the **Avalanche** won"

$\models$

"**Either** the Broncos won **or** the Avalanche won"

Entailment

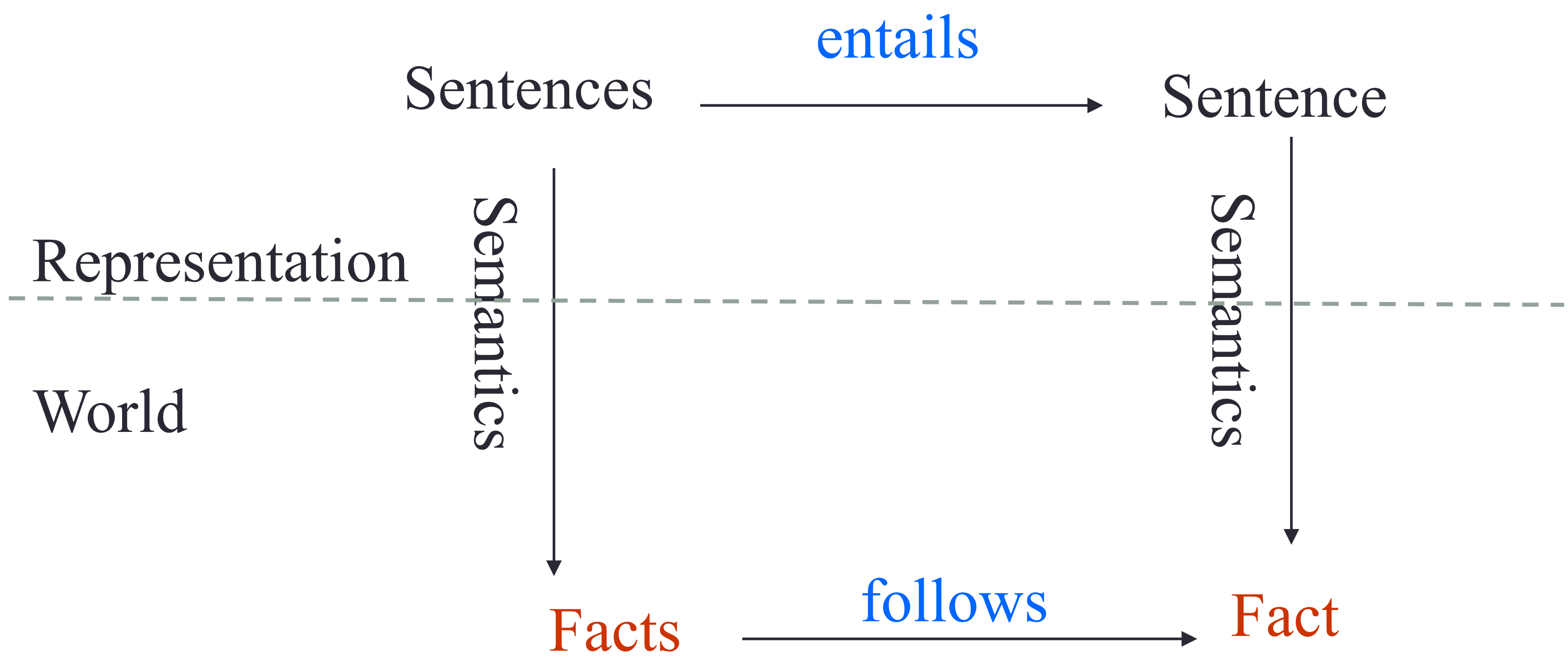
- It means that one thing follows from another:

$$KB \models \alpha$$

- Knowledge base *KB* entails sentence α if and only if α is true in all worlds where *KB* is true. (Example: s = There is no Pit in [2,1])
  - E.g., the KB containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"
  - E.g., x+y = 4 entails 4 = x+y
  - Entailment is a relationship between sentences that is based on semantics

Entailment is different than inference

Entailment vs. Inference



<u>Example</u>: Sentence (A) *The president was assassinated.* entails (B) *The president is dead.* Here

1- (A) being true forces (B) to be true, or, equivalently, that (B) being false forces (A) to be false.

2- With inference you draw a conclusion. With entailment or implication the conclusion is pushed from the premise to the level of fact or result.

In other words, inference draws a **conclusion**, whereas entailment draws a **result**.

Entailment VS Inference

Entailment:

It is a Relation Between 2 Sentences where one Logically implies the other. If A is true then B is true too.

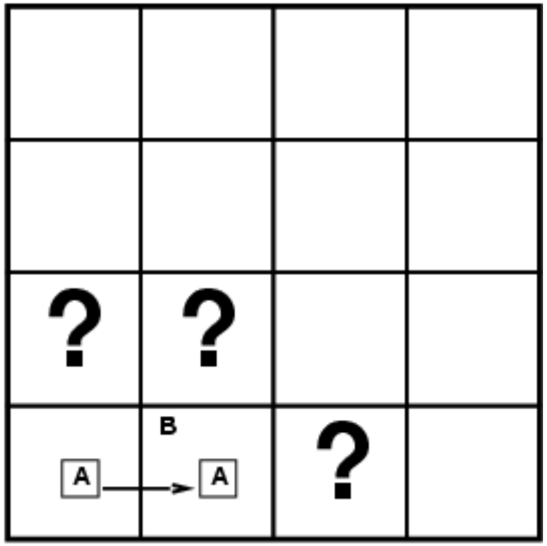
Inference:

It is the Process of drawing a Conclusion based on evidence or Reasoning. "Socrates is mortal" & "John is a man" we can infer that "John is Also Mortal" Based on the fact that all men are Mortal.

Requires Prior History.

Entailment in the Wumpus World

- Situation after detecting nothing in [1,1], moving right,
- Consider possible models for *KB* assuming only pits

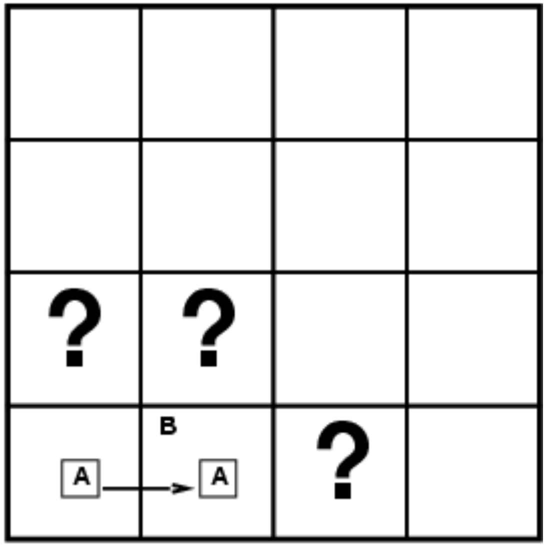


Definition

Number of possible models when placing pits in three squares:

Entailment in the Wumpus World

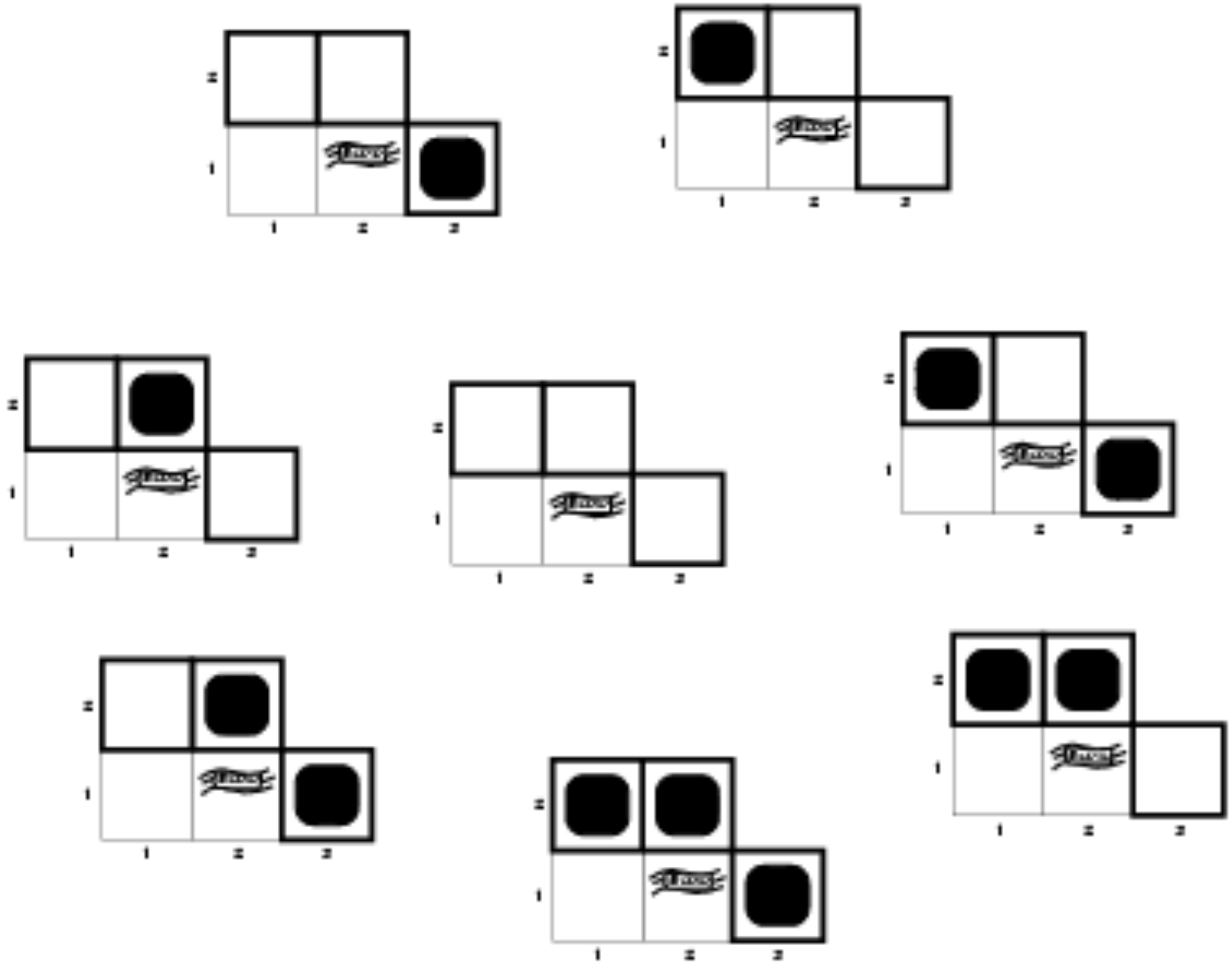
- Situation after detecting nothing in [1,1], moving right,
- Consider possible models for *KB* assuming only pits



Definition

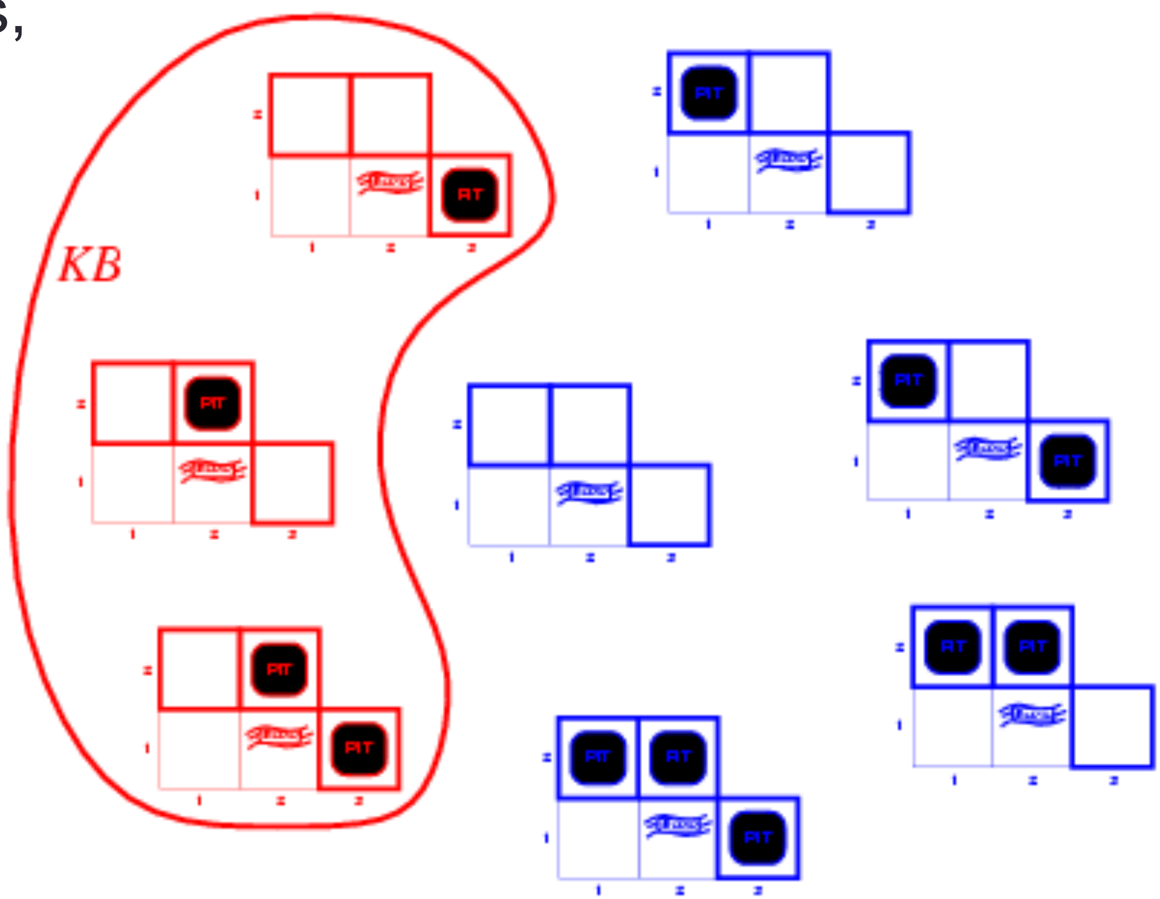
Number of possible models when placing pits in three squares: $2^3 = 8$

Entailment in the Wumpus World



Entailment in the Wumpus World

- *KB* of wumpus-world = rules + observations,
-

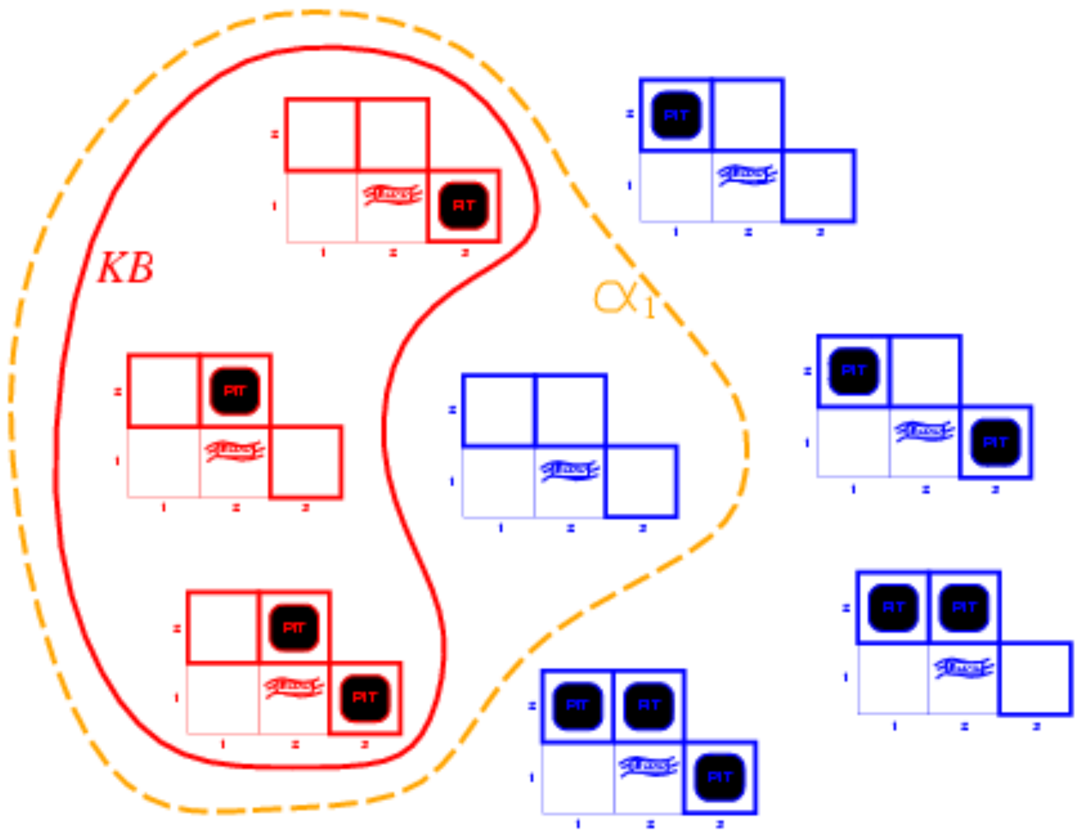


Entailment in the Wumpus World

- *KB* of wumpus-world = rules + observations,

- *α1 = "[2,1] is safe"*,

KB $\models \alpha_1$,



Entailment in the Wumpus World

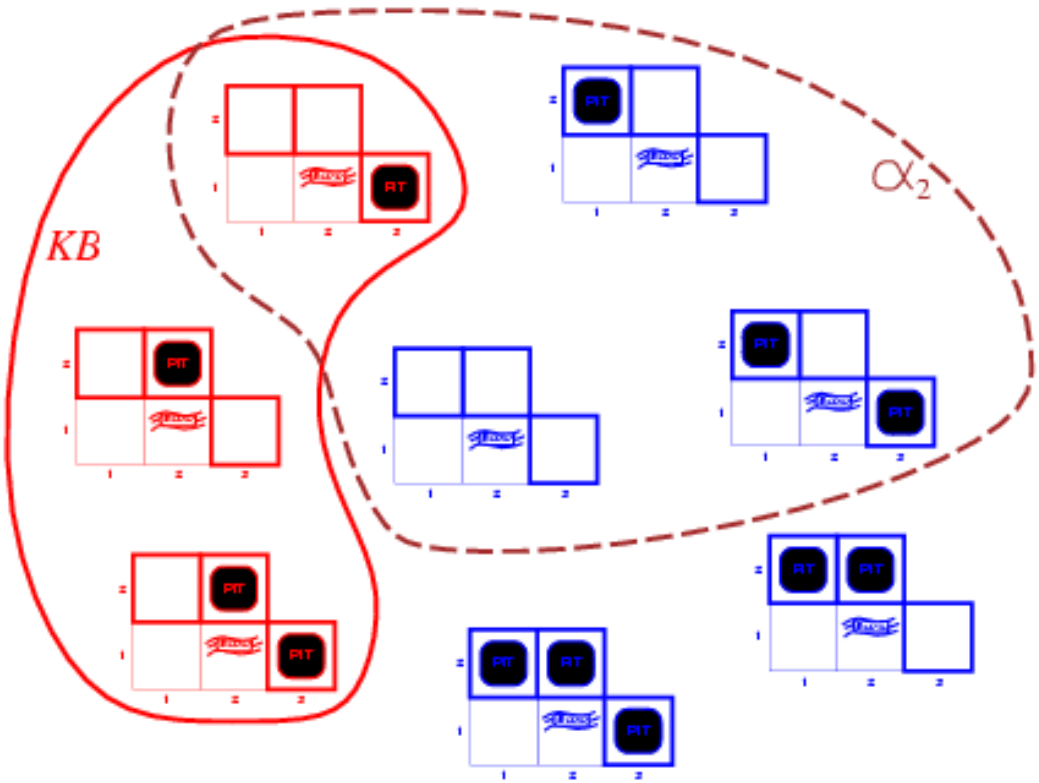
Knowlege Base

- *KB* of wumpus-world = rules + observations,

- *α1 = "[1,2] is safe"*, $KB \models \alpha_1$,

- α_2 = "[2,2] is safe", $KB \not\models \alpha_2$

-



Propositional Logic

A very simple logic

Lecture III

Proposition Logic

very general.

Definition of Statement

Something objective.

A statement can be defined as a declarative sentence, or part of a sentence, that is capable of having a truth-value, such as being true or false.

Example of Statement

- **Obama is the 44th President of the United States**
- **Paris is the capital of France**

Not up for Debate

FAX

Example of Statement

- **Either Ganymede is a moon of Jupiter or Ganymede is a moon of Saturn** *

Proposition Logic

Definition

modyfiing sentences to make more complex compound

Propositional logic, also known as *sentential logic* or *statement logic*, is the branch of logic that studies ways of joining and/or modifying entire propositions, statements or sentences to form more complicated propositions, statements or sentences.*

Propositional logic can be connected using *Boolean connectives* to generate sentences with more complex meanings

Proposition Logic Syntax

Basics

- Symbols look like P, Q, R, etc.
- Connectives look like:
  - ¬ negation, "not"
  - ^ conjunction, "and"
  - V disjunction, "or"
  - => Implication, "implies"
  - ⇔ biconditional, "equivalent"

Proposition Logic Syntax

Example

P	"P is true"	
¬P	"P is false"	negation
P V Q	"either P is true or Q is true or both"	disjunction
P ^ Q	"both P and Q are true"	conjunction
P => Q	"if P is true, the Q is true"	implication
P ⇔ Q	"P and Q are either both true or both false"	equivalence

Proposition Logic Syntax

Basics

- **The Proposition symbols S_1, S_2 are sentences:**

  - If S is a sentence, $\neg S$ is a sentence (negation)
  - If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)
  - If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)
  - If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)
  - If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Proposition Logic Semantics

- Specifies how to compute the truth value of any sentence
- Truth Table for logical Connectives

Truth Table

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Evaluating Complex Expressions

Given

T F T

$P = true \quad Q = false \quad R = true$

Evaluate

$P \lor R \Rightarrow \neg(Q \land \neg R)$

(1) true ∨ true => ¬ (false ^ ¬ true)
(2) true ∨ true => ¬ (false ^ false)
(3) true ∨ true => ¬ (false)
(4) true ∨ true => true
(5) true => true
(6) true

A simple knowledge base..A simple inference procedure

- Truth tables can be used not only to define the connectives, but also to test for validity: How to test the validity?
  - If a sentence is true in every row, it is valid. 1
  - A truth table for “Premises imply Conclusion” 2
  - A simple knowledge base for Wumpus (P246) 3

A simple knowledge base

$P_{x,y}$ is true if there is a pit in $[x, y]$.
$W_{x,y}$ is true if there is a wumpus in $[x, y]$, dead or alive.
$B_{x,y}$ $B_{y,y}$ is true if the agent perceives a breeze in $[x, y]$.
$S_{x,y}$ is true if the agent perceives a stench in $[x, y]$.

The sentences we write will suffice to derive $\neg P_{1,2}$ (there is no pit in [1,2]), as was done informally in Section 7.3, We label each sentence R_i so that we can refer to them:

- There is no pit in [1,1]:

  R_1 : $-P_{1,1}$

- A square is breezy if and only if there is a pit in a neighboring square. This has to be stated for each square; for now, we include just the relevant squares:

  R2 : B1,1 (P1,2 V .P2,1) .

  R_3 : B2,1 tZ (Pit_ V .P2,2 V P3,1) -

- The preceding sentences are true in all wumpus worlds. Now we include the breeze percepts for the first two squares visited in the specific world the agent is in, leading up to the situation in Figure 7.3(b).

  R4 : ı$B_{1,1}$.

  R5 : B2,1 •

A simple inference procedure

Byo?

B1,1	B2,1	P1,1	$P_{1,2}$	P2,1	P2,2	$P_{3,1}$	R1	R2	R_3	R4	R5	KB
false *false* : *false*	*false* *false* *true*	*false* *false* . *false*	*false* *false* . *false*	*false* *false* *false*	*false* false : *false*	*false* *true* *false*	*true* *true* : *true*	*true* *true* : *true*	*true* *false* : *false*	*tree* *tree* : *true*	*false* *false* : *true*	*false* *false* : *false*
false *false* *false*	[illegible] [illegible] [illegible]	*false* *false* *false*	*false* *false* *false*	*false* *false* *false*	*false* *true* *true*	*true* *false* *true*	*true* true *true*	e	*true* true *true*			
false : *true*	*true* : *true*	*false* : *true*	*false* *true*	*true* : true	*false* : *true*	*false* : *true*	*true* : *false*	*false* : *true*	*false* : *true*	*true* : *false*	*true* : *true*	*false* : *false*

A simple inference procedure

```
function TT-ENTAILS?(KB, α) returns true or false
   inputs: KB, the knowledge base, a sentence in propositional logic
           α, the query, a sentence in propositional logic

   symbols ← a list of the proposition symbols in KB and α
   return TT-CHECK-ALL( KB , α, symbols, {})
-----------------------------------------------------------------------
function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
   if EMPTY?(symbols) then
       if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
       else return true // when KB is false, always return true
   else do
       P ← FIRST(symbols)
       rest ← REST(symbols)
       return (TT-CHECK-ALL(KB, α, rest, model U {P = true})
                and
               TT-CHECK-ALL(KB, α, rest, model U {P = false }))
```

For more details, study the Section 7.4

Other Logics Branches

good for Complexity

- Propositional logic generates sentences with more complex meanings, but does not specify how objects are represented.

represent using objects

- First order logic represents worlds using objects and predicates on objects with connectives and quantifiers.
- Temporal logic assumes that the world is ordered by a set of time points or intervals and includes mechanisms for reasoning about time. Where time is the Important element.

First-Order Logic

In which we notice that the world is blessed with many objects, some of which are
related to other objects, and in which we endeavor to reason about them.

Lecture IV

Pros and Cons of Propositional Logic

1 • Propositional logic is declarative: pieces of syntax correspond to facts Con ?

2 • Propositional logic allows for partial / disjunctive / negated information (unlike most data structures and DB) Pro

3 • Propositional logic is compositional: the meaning of $B_{11} \wedge P_{12}$ is derived from the meaning of B_{11} and P_{12}

4 • Meaning of propositional logic is context independent: (unlike natural language, where the meaning depends on the context)

5 • Propositional logic has very limited expressive power: (unlike natural language) con
  - E.g. cannot say Pits cause Breezes in adjacent squares except by writing one sentence for each square

Propositional Logic

Pros	Cons
- Propositional Logic is Declaritive.	- Has very Little expressive Power.
- Context independent.	- Propositional logic is Compositional.

First-Order Logic

- AKA First-Order Predicate Logic
- AKA First-Order Predicate Calculus

- It uses quantified variables over non-logical objects and allows the use of sentences that contain variables.

- Rather than proposition: “Socrates is a man” one can have expressions in the form "there exists X such that X is Socrates and X is a man" and there exists is a quantifier while X is a variable.

- Propositional logic which does not use quantifiers or relations

First-Order Logic

Propositional logic

- AKA First-Order Predicate Logic
- AKA First-Order Predicate Calculus

- Much more powerful than the propositional (Boolean) logic
  - Greater expressive power than propositional logic
    - We no longer need a separate rule for each square to say which other squares are breezy/pits
  - Allows for facts, objects, and relations
    - In programming terms, allows classes, functions and variables

Pros of First-Order Logic

- First-Order Logic assumes that the world contains:
  - Objects
    - E.g. people, houses, numbers, theories, colors, football games, wars, centuries, …
  - Relations
    - E.g. red, round, prime, bogus, multistoried, brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, …
  - Functions
    - E.g. father of, best friend, third quarter of, one more than, beginning of, …

The language of first-order logic is built around objects and relations.

Syntax of First-Order Logic

• Constants	KingJohn, 2, …
• Predicates	Brother, >, …
• Functions	Sqrt, LeftArmOf, …
• Variables	x, y, a, b, …
• Connectives	$\wedge \vee \neg \Rightarrow \Leftrightarrow$
• Equality	=
• Quantifiers	$\exists \forall$

Components of First-Order Logic

- Term
  - Constant, e.g. Red
  - Function of constant, e.g. Color(Block1)

- Atomic Sentence
  - Predicate relating objects (no variable)
    - Brother (John, Richard)
    - Married (Mother (John), Father (John))

- Complex Sentences
  - Atomic sentences + logical connectives
    - Brother (John, Richard) $\wedge\neg$ Brother (John, Father (John))

Components of First-Order Logic

- Quantifiers Temorary variable?
  - Each quantifier defines a variable for the duration of the following expression, and indicates the truth of the expression…
- Universal quantifier "for all" ∀
  - The expression is true for every possible value of the variable
- Existential quantifier "there exists" ∃
  - The expression is true for at least one value of the variable

Deducing Hidden Properties

- **Definition for the Breezy predicate:**
  - If a square is breezy, some adjacent square must contain a pit
    - $\forall y \text{ Breezy}(y) \Rightarrow \exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$
  - If a square is not breezy, no adjacent pit contains a pit
    - $\forall y \neg\text{Breezy}(y) \Rightarrow \neg\exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$
  - **Combining these two…**
    - $\forall y \text{ Breezy}(y) \Leftrightarrow \exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$

Keeping Track of Change

- Often, facts hold in situations, rather than eternally
  - E.g. Holding(Gold, now) rather than just Holding (Gold)

- Situation calculus is one way to represent change in FOL:
  - Adds a situation argument to each non-eternal predicate
    - E.g. Now in Holding(Gold, Now) denotes a situation


Any questions?