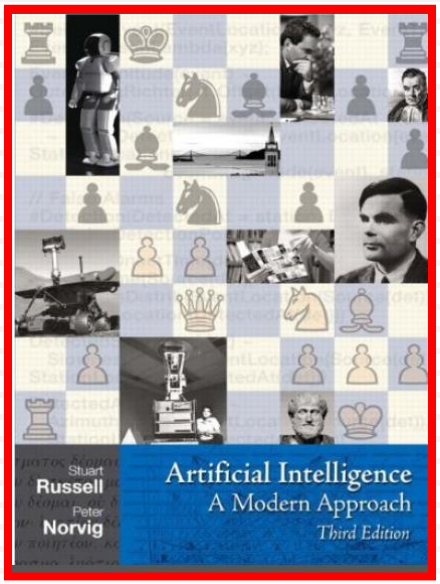
Princess Nora University
Faculty of Computer & Information Systems

جامعة الأميرة نورة بنت عبد الرحمن
Princess Nora Bint Abdul Rahman University

ARTIFICIAL INTELLIGENCE

(CS 370D)

جامعة الأميرة نورة بنت عبد الرحمن
Princess Nora Bint Abdul Rahman University



(CHAPTER-7)
LOGICAL AGENTS

Outline

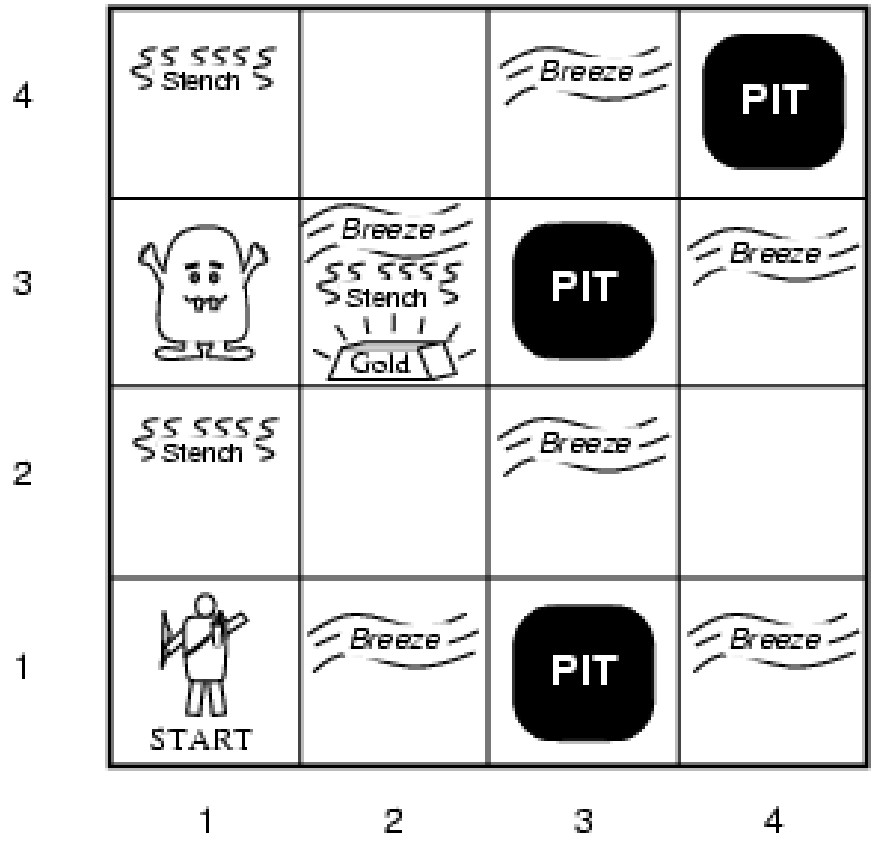
- Agent Case (Wumpus world)
- Knowledge-Representation
- Logic in general - models and entailment
- Propositional (Boolean) logic

Dr. Abeer Mahmoud
(course coordinator)

Agent Case (Wumpus world)

Dr. Abeer Mahmoud
(course coordinator)

The Wumpus World



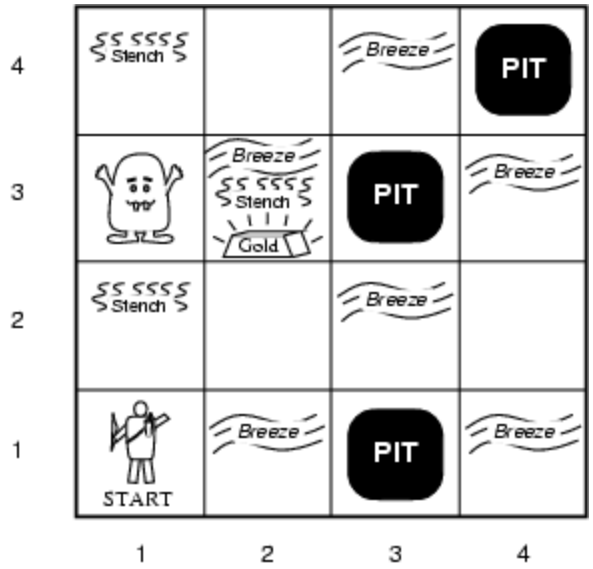
- ✓ The **wumpus** world is a cave consisting of rooms connected by passageways.
- ✓ Lurking somewhere in the cave is the terrible wmnpus, a beast that eats anyone who enters its room.
- ✓ The wumpus can be shot by an agent, but the agent has only one arrow.
- ✓ Some rooms contain bottomless pits (very deep hole) that will trap anyone who wanders into these rooms (except for the wumpus, which is too big to fall in).

Dr. Abeer Mahmoud
(course coordinator)

Wumpus World PEAS description

- **Performance measure**
  - Gold→ +1000, death →-1000
  - -1 per step, -10 for using the arrow

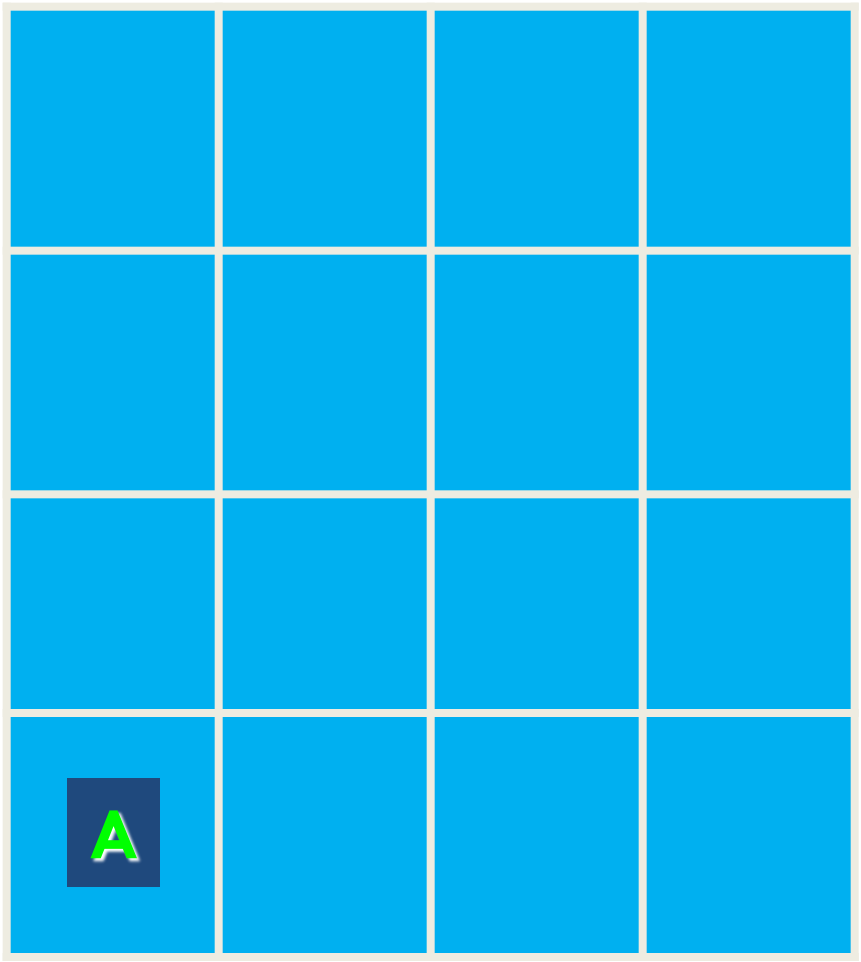
- **Environment**
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy(windy)
  - Glitter **iff** gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops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



- **Sensors**: Stench, Breeze (for movement), Glitter (gold), Bump (hit), Scream

- **Actuators**: Left turn, Right turn, Forward, Grab, Release, Shoot

Exploring Wumpus World



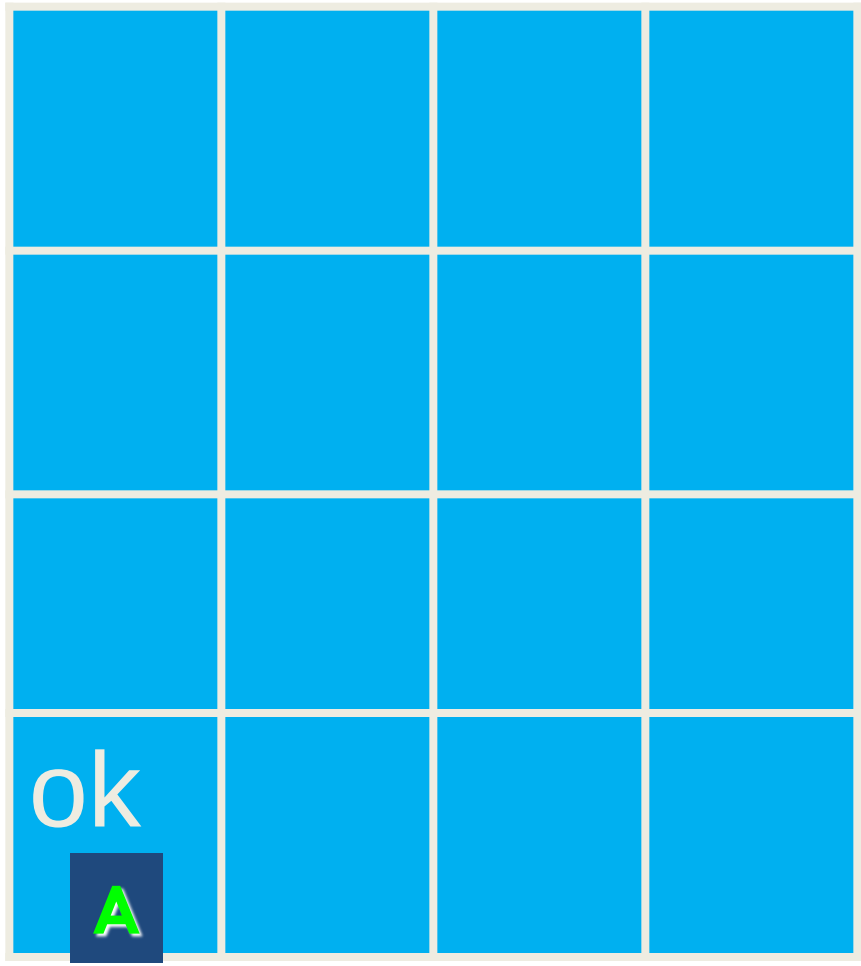
Dr. Abeer Mahmoud
(course coordinator)

Exploring Wumpus World

Ok because:

Haven't fallen into a pit.

Haven't been eaten by a Wumpus.



Dr. Abeer Mahmoud
(course coordinator)

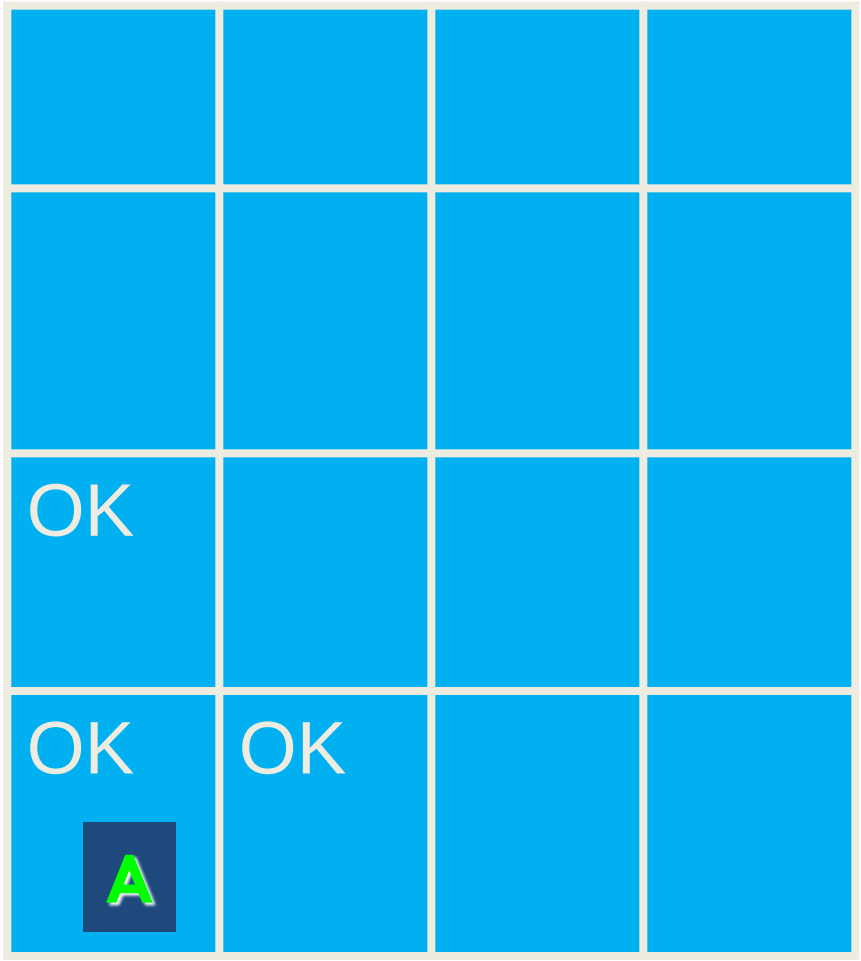
Exploring Wumpus World

OK since

no Stench,

no Breeze,

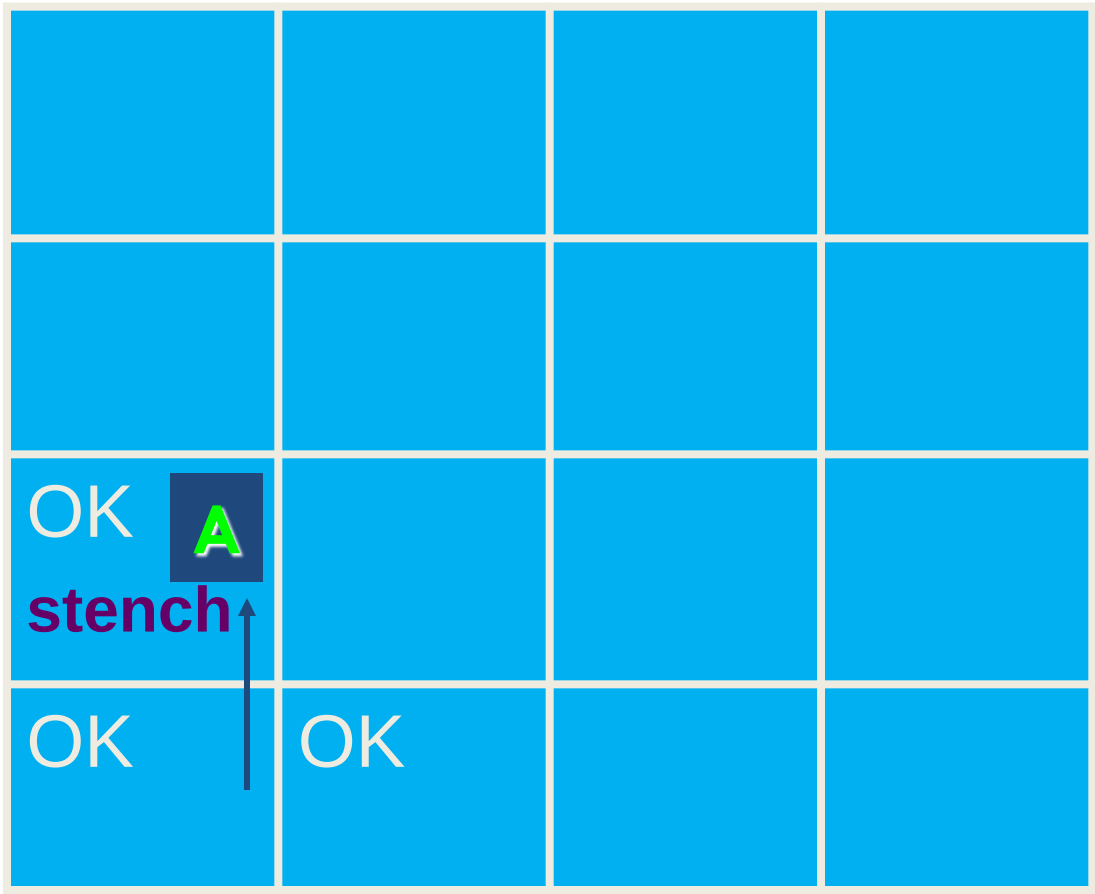
neighbors are safe (OK).



Dr. Abeer Mahmoud
(course coordinator)

Exploring Wumpus World

We move and smell a stench.



Dr. Abeer Mahmoud
(course coordinator)

Knoweldge Representation & Reasoning

Exploring Wumpus World

We can infer the following.

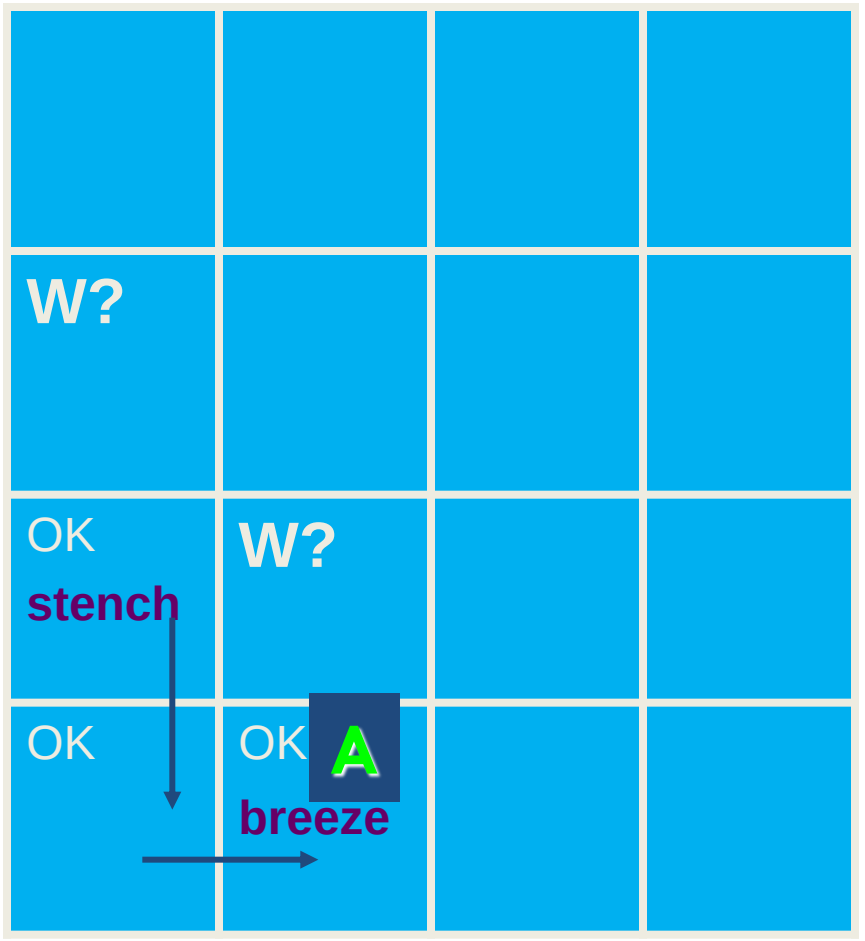
Note: **square (1,1) remains OK.**

Dr. Abeer Mahmoud
(course coordinator)

Exploring Wumpus World

Move and feel a breeze

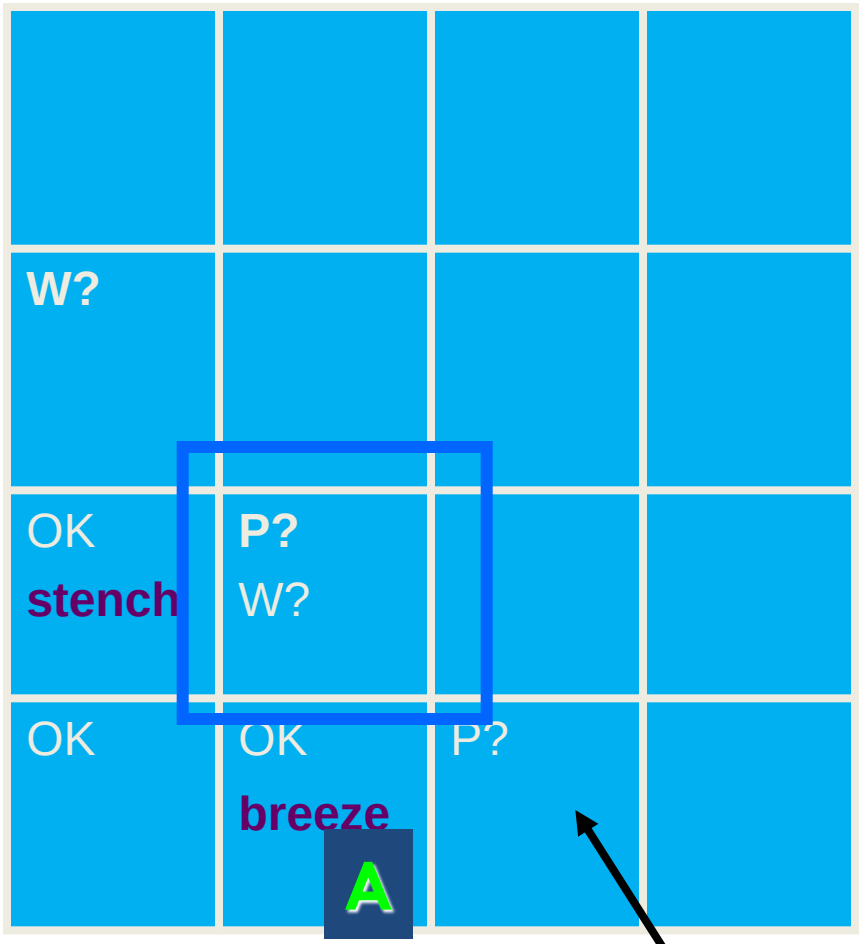
What can we conclude?



Dr. Abeer Mahmoud
(course coordinator)

Exploring Wumpus World

But, can the 2,2 square really have either a Wumpus or a pit?

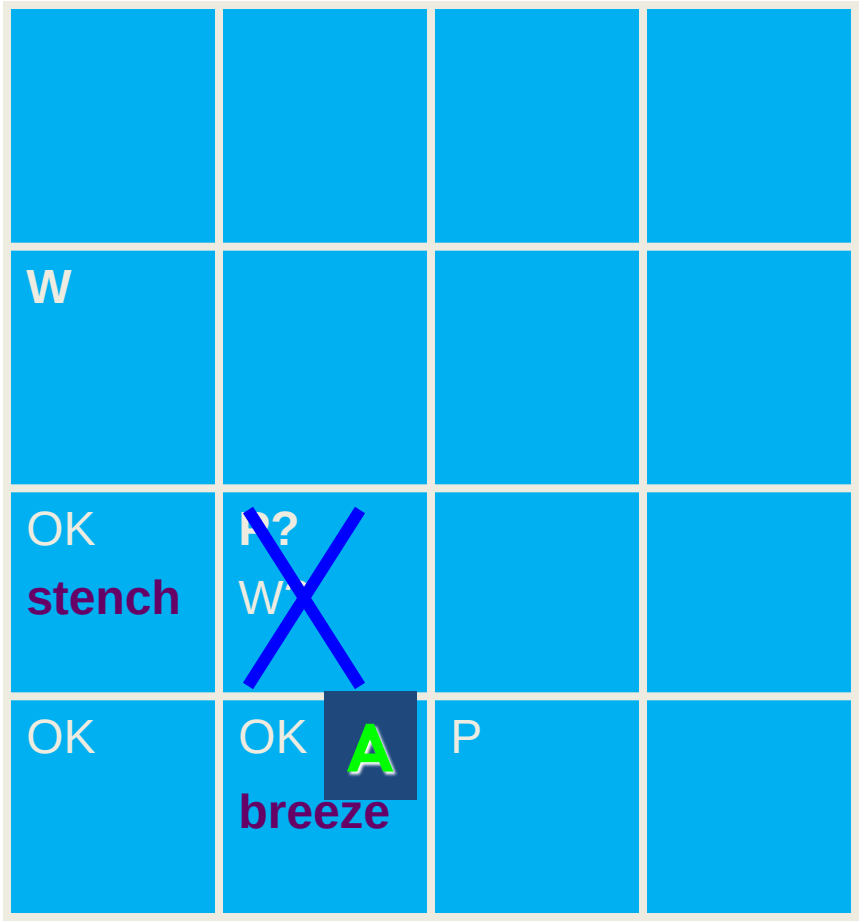


NO!

And what about the other P? and W? squares

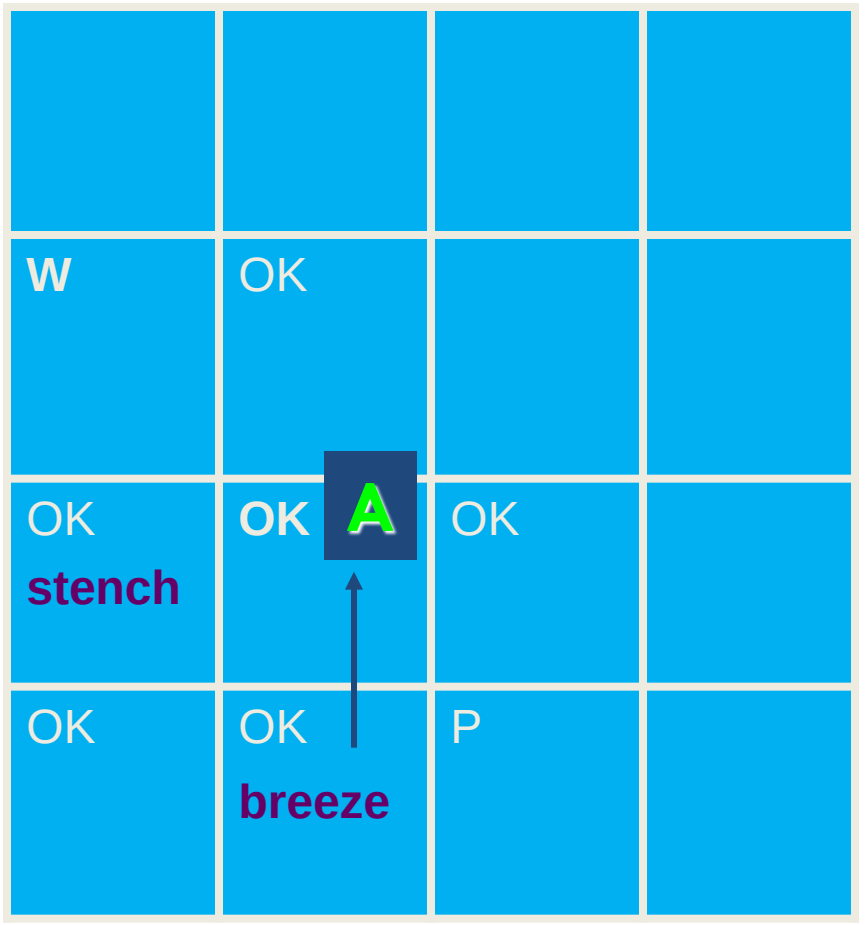
13

Exploring Wumpus World



Dr. Abeer Mahmoud
(course coordinator)

Exploring Wumpus World



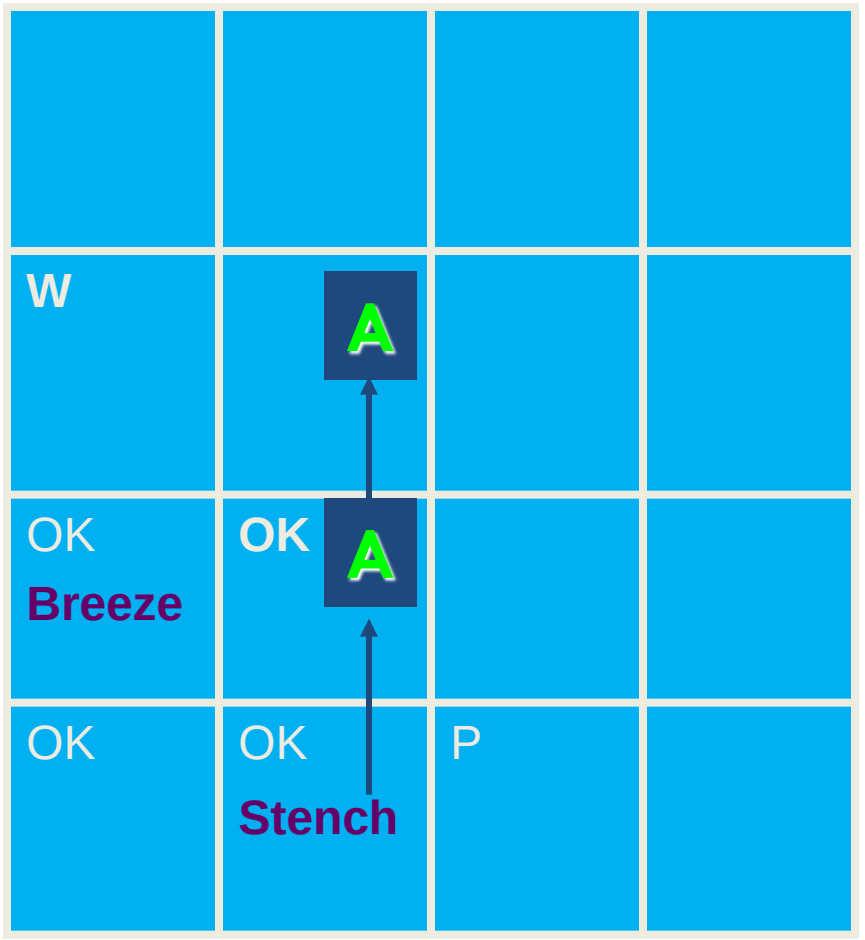
Dr. Abeer Mahmoud
(course coordinator)

Exploring Wumpus World

…

And the exploration continues onward until the gold is found.

…



Dr. Abeer Mahmoud
(course coordinator)

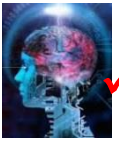
What does an agent need to perform well in the wumpus world?

- It needs to represent the knowledge about the wumpus world: ex, stench → wumpus nearby.
- It needs to incorporate new knowledge that it discovers when exploring the cave →feeling a breeze in the current square.
- Deduce from the preceding information the appropriate actions (safe and rewarding) :
- The agent needs a **knowledge-base.**

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

✓ **First**, it TELLS the knowledge base what it perceives.

✓ **Second**, it ASKS the **knowledge base** what action it should perform



✓ **Third**, the agent program TELLS the knowledge base which action was chosen, and the agent executes the action.

Dr. Abeer Mahmoud
(course coordinator)

Knowledge bases

- **Knowledge base**: set of **sentences**. Each **sentence** is expressed in a language called a **knowledge representation language.**

- **Sentence**: a sentence represents some assertion about the world.

- **Inference**: Process of deriving new sentences from old ones.

Dr. Abeer Mahmoud
(course coordinator)

Types of Knowledge

Declarative knowledge	Concepts Facts Objects	Describes **what** is known about a problem. This includes simple statements that are asserted to be either **true** or **false**. This also includes a list of statements that more fully describes some **object** or **concept** (object-attribute-value triplet).

Dr. Abeer Mahmoud
(course coordinator)

Types of Knowledge

Procedural knowledge	Rules Strategies Agendas Procedures	Describes **how** a problem is solved. This type of knowledge provides direction on how to do something.

Dr. Abeer Mahmoud
(course coordinator)

Some General Knowledge Representations

1. Logical Representations
2. Production Rules
3. Semantic Networks
   - Conceptual graphs, frames
4. *Description Logics* (not covered in this course)

Dr. Abeer Mahmoud
(course coordinator)

1-Logical Representation

Dr. Abeer Mahmoud
(course coordinator)

1-Logical Representation

- A language with concrete rules
- Many ways to translate between languages
  - A statement can be represented in different logics
  - And perhaps differently in same logic
- Not to be confused with logical reasoning
  - Logics are languages, reasoning is a process (may **use** logic)

Dr. Abeer Mahmoud
(course coordinator)

1-Logical Representation

Syntax and Semantics

- **Syntax**
  - Rules for constructing legal sentences in the logic
  - Which symbols we can use (English: letters, punctuation)
  - How we are allowed to combine symbols
- **Semantics**
  - How we interpret (read) sentences in the logic
  - Assigns a meaning to each sentence

- Example: "All lecturers are seven foot tall"
  - A valid sentence (syntax)
  - And we can understand the meaning (semantics)
  - This sentence happens to be false (there is a counter example)

Dr. Abeer Mahmoud
(course coordinator)

Logic

- Oldest form of knowledge representation in a computer is logic
- Logic is concerned with the truthfulness of a chain of statements.
- An argument is true if and only if, when all assumptions are true, then all conclusions are also true.
- **Two kinds of logic:**
  1. **Propositional Logic**
  2. **Predicate Calculus**

- Both use symbols to represent knowledge and operators applied to the symbols to produce logical reasoning

Dr. Abeer Mahmoud
(course coordinator)

1.Propositional Logic (PL)

- Propositional logic represents and reasons with propositions.
- P.L. assigns symbolic variable to a proposition such as
  - A = The car will start
- In. P. L. if we are concern with the **truth** of the statement, we will check the truth of A.

Dr. Abeer Mahmoud
(course coordinator)

1.Propositional Logic (PL)

Operators	Symbol
AND	$\wedge$, &, $\cap$
OR	$\vee$, $\cup$, +
NOT	$\neg$, ~
IMPLIES	$\supset$, $\rightarrow$
EQUIVALENT	$\equiv$

1.Propositional Logic (PL)

- Propositions that are linked together with connectives, such as **AND**, **OR**, **NOT**, **IMPLIES**, and **EQUIVALENT**, are called **compound statements.**

- Example:

IF	The Students Work Hard	→	A	
AND	Always come to lectures	→	B	
AND	Do all their homework's	→	C	
THEN	they will get an A	→	D	

- Using the symbols: $A \wedge B \wedge C \rightarrow D$

Dr. Abeer Mahmoud
(course coordinator)

1.Propositional Logic (PL)

Truth Table

A	B	A and B	A or B	Not A	A ≡ B
F	F	F	F	T	T
F	T	F	T	T	F
T	F	F	T	F	F
T	T	T	T	F	T

Dr. Abeer Mahmoud
(course coordinator)

1.Propositional Logic (PL)

- **Implies Operator:** $C = A \rightarrow B$
- For implication C, if A is true, then B is implied to be true
- The implies return a F when A is TRUE and B is FALSE Otherwise it return a TRUE.

A	B	C
F	F	T
F	T	T
T	F	F
T	T	T

Ex : A = it is raining
B= I have an umbrella

Ex : A = program code
B= output

1.Propositional Logic (PL)

Idempotent Laws	$A \rightarrow B \equiv \neg A \cup B$ $A \cap \neg A \equiv F$ $A \cup \neg A \equiv T$
Commutative Laws	$A \cap B \equiv B \cap A$ $A \cup B \equiv B \cup A$
Distributive Laws	$A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$
Associative Laws	$A \cap (B \cap C) \equiv (A \cap B) \cap C$ $A \cup (B \cup C) \equiv (A \cup B) \cup C$
Absorptive Laws	$A \cup (A \cap B) \equiv A$ $A \cap (A \cup B) \equiv A$
DeMorgan's Laws	$\neg(A \cap B) \equiv \neg A \cup \neg B$ $\neg(A \cup B) \equiv \neg A \cap \neg B$

Dr. Abeer Mahmoud
(course coordinator)

1.Propositional Logic (PL)

- **P.L. offers techniques for capturing facts or rules in symbolic form and then operates on them through use of logical operators.**
- PL provide method of managing statements that are either TRUE or FALSE.
- Prolog is based on PC

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- Some P.L. weakness:

1. Limited ability to express knowledge and lose much of their meanings.
2. Not all statements can be represented.

All *men are mortals.*

Some *dogs like cats.*

- Thus, need a more general form of logic capable of representing the details.

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- Propositional logic combines atoms
  - An atom contains no propositional connectives
  - Have no structure (today_is_wet, john_likes_apples)

- **Predicates** allow us to talk about objects
  - Properties: is_wet(today)
  - Relations: likes(john, apples)
  - True or false

- In **predicate logic** each atom is a predicate
  - e.g. first order logic, higher-order logic

Dr. Abeer Mahmoud
(course coordinator)

First Order Logic

- More expressive logic than propositional
- Enhances processing by allowing the use of variables and functions.

Use symbols that represent

constants are objects: john, apples

predicates

are properties and relations:

likes(john, apples)

variables

represent any object: likes(X, apples)

functions

transform objects:

likes(john, fruit_of(apple_tree))

- Operate on these symbols using PL operators.

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- **Constant**

- Specific objects or properties about a problem.
- Begin with lower case.
- Example: ahmad, elephant and temperature
- ahmad represent object Ahmad. Can also use A or X instead of Ahmad.

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- **Predicates**

- Divide proposition into 2 parts:
- predicate: assertion about object
- argument: represent the object
- Example: To represent Azizi teach Expert system(ES),
  - teach (azizi, ES)
  - teach is a *predicate*, denoting relationship between arguments. The 1st letter must be in lower case.

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- **Variables**

- Represent general classes of objects or properties
- Written as symbols beginning with upper case.
- To capture the proposition Azizi teach ES, we write:
  - **teach (X,Y)**
  - **X = Azizi and Y = ES**

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- **Functions**

- Permits symbol to be used to represent function.
- A function denotes a mapping from entities of a set to a unique element of another set.
  - **father(azizi) = zakaria mother(azizi) = zaharah.**
  - **Can be also used within predicates. For example:**
  - **husband (father(azizi), mother(azizi)) = husband(zakaria,zaharah)**

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- **Operations**
- PC uses the same operators found in P.L.
- Proposition: John likes Mary likes(john,mary)
- Bob likes Mary likes(bob,mary)
- 2 persons like Mary. To account for jealousy:
  - **likes (X,Y) AND likes(Z,Y) implies NOT likes (X,Z)**

or

  - **likes (X,Y) ∧ likes (Z,Y) → ¬likes(X,Z)**

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- P.C. introduce 2 symbols called variable quantifiers.

1. ∀ universal quantifier: **for all.**
2. ∃ existential quantifier: **there exist**

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- ∀ Indicates the expression is TRUE for all values of designated variable.
- Example:
  - ∀ X likes (X,mary)
  - means for all values of X, the statement is true, everybody likes Mary

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- ∃ indicates the expression is TRUE for some values of the variable; at least one value exist that makes the statement true:
- ∃ X likes (X,mary)

Dr. Abeer Mahmoud
(course coordinator)

2.Predicate Calculus (PC)

- Parentheses are used to indicate the scope of quantification
- $\forall$ X (likes(X,mary) $\wedge$ nice(mary) $\rightarrow$ nice (X))
  - determines all instances of X who like Mary and if Mary is nice, then it is implied that those who like Mary are also nice.

Dr. Abeer Mahmoud
(course coordinator)

Reasoning with logic

- PC can provide reasoning capability to intelligent system
- Reasoning requires the ability to infer conclusions from available facts.
- One simple form of inference is modus ponen:
  - **IF** A is true

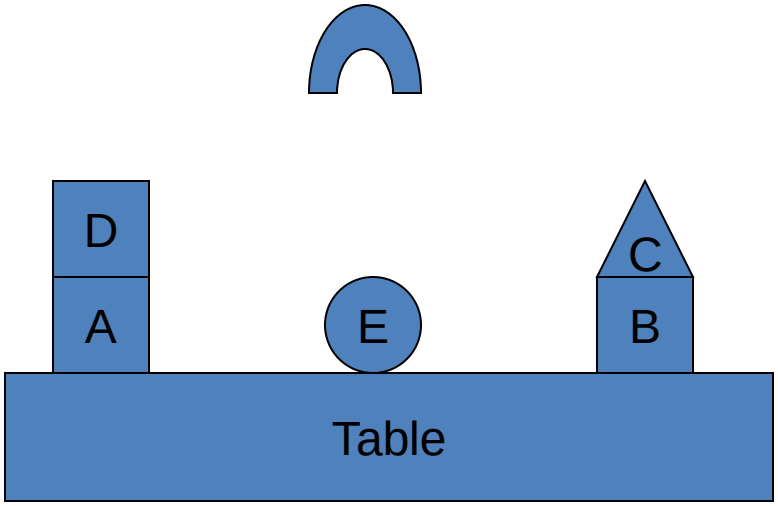
    AND A $\rightarrow$ B is true

    THEN B is true

Dr. Abeer Mahmoud
(course coordinator)

Reasoning with logic

- Robot Control Example
- The function of the robot is move a specified block to some specified location



Dr. Abeer Mahmoud
(course coordinator)

Reasoning with logic

Robot Control Example

- Description of the block world using PC using the following logical assertions:

1. **cube(a), cube(b), cube(d), pyramid(c), sphere(e), hand(hand), table (table1)**
2. **on(a,table1), on(b,table1), on(d,a), on(c,b)**
3. **holding(hand,nothing)**

Dr. Abeer Mahmoud
(course coordinator)

Reasoning with logic

- Robot Control Example
- The goal might be to put some block on other block, for example put block **b** onto block **a**:

  put_on(b,a)

- To accomplish this the robot need to obtain block **b** and make certain that block **a** is clear:

  hand_holding(b) ∧ clear(a) → put_on(b,a)

- To move any block, in variable form:

  ∀X ∃Y (hand_holding(X) ∧ clear(Y) → put_on(X,Y))

where X is the block to be move and Y is the target block

Dr. Abeer Mahmoud
(course coordinator)

Reasoning with logic

- Robot Control Example

- One of the robot task when instructed to pick up and move some blocks is to determine if it is clear.

- If not clear, need to remove any item on the block:

$$\forall X (\neg \exists Y \ on(Y,X) \rightarrow clear(X))$$

For all X, X is clear if there does not exist a Y such that Y is on X, would produce the following assertions:

clear(c), clear(d)

Dr. Abeer Mahmoud
(course coordinator)

Logic is a Good Representation

- Fairly easy to do the translation when possible
- Branches of mathematics devoted to it
- It enables us to do logical reasoning
  - Tools and techniques come for free
- Basis for programming languages
  - Prolog uses logic programs (a subset of FOL)

Dr. Abeer Mahmoud
(course coordinator)

Thank you

End of Chapter 7-part1

Dr. Abeer Mahmoud
(course coordinator)