CS 461
Artificial Intelligence

Dr. Hashim Yasin

Knowledge, Reasoning and Logic

Knowledge

Humans know things …

$\Rightarrow$ **the knowledge helps them to do various tasks.**

$\Rightarrow$ **The knowledge has been achieved**

- not by purely reflex mechanisms
- but by the processes of **reasoning**

- In AI, the example is **knowladge-based agent** which contains **set of sentences** referred as **knowledge-base.**

Knowledge-based Agent

For a generic knowledge-based agent:

- A **percept is given** to the agent.
- The agent **adds the percept** to its knowledge base.
- **Perform best action** according to the knowledge base.
- Tells the knowledge base that it has in fact **taken that action**.

Knowledge-based Agent

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

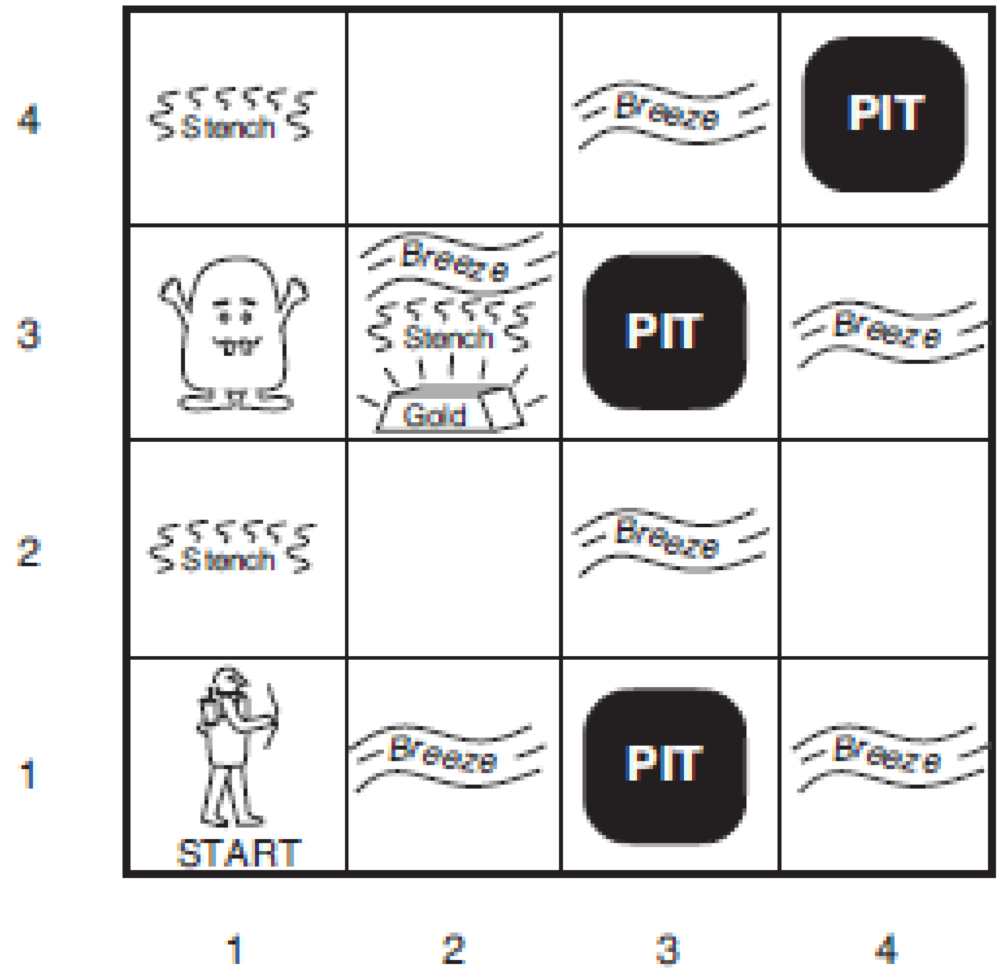
  TELL(KB, MAKE-PERCEPT-SENTENCE(*percept*, t))
  action ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(*action*, t))
  $t \leftarrow t + 1$
  return *action*

constructs a **sentence** asserting that the agent ***perceived the given percept*** at time ***t***

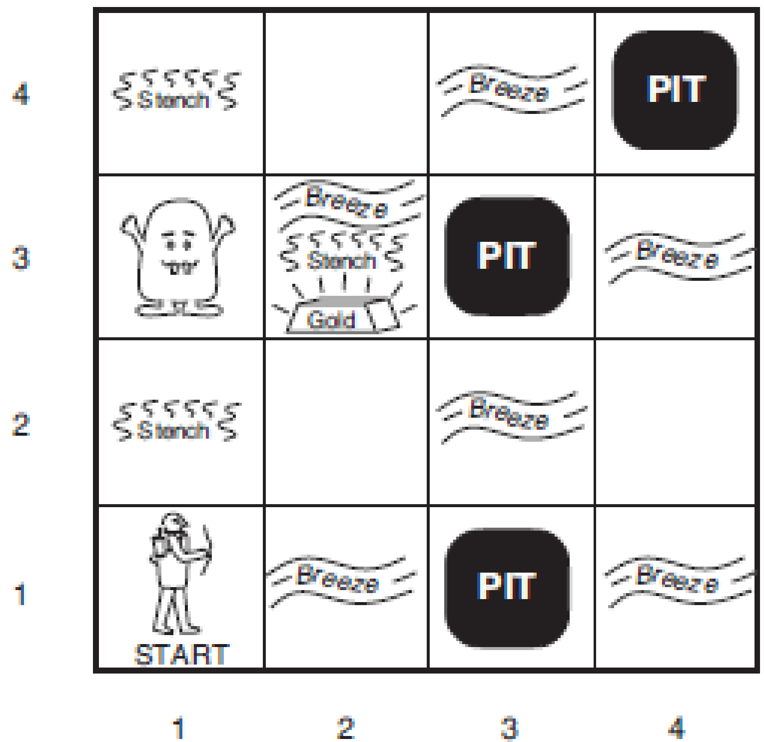
constructs a sentence that asks ***what action should be done*** at time ***t***

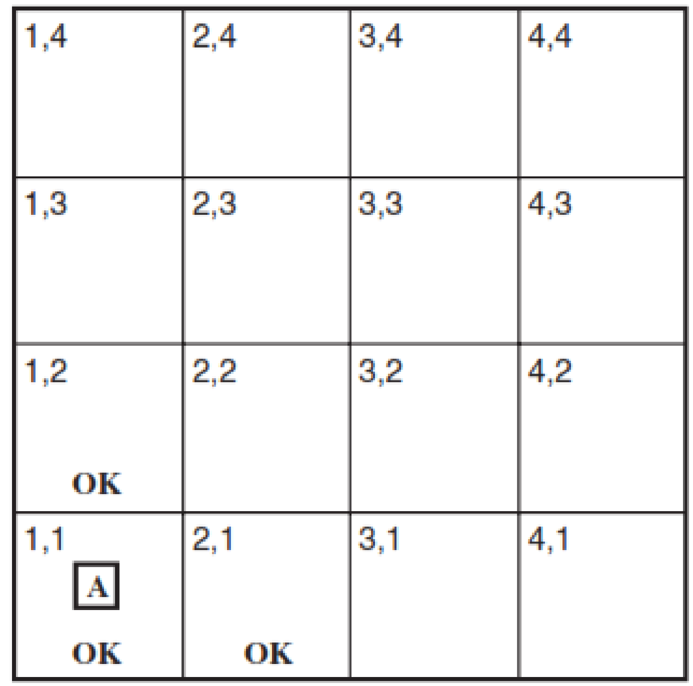
constructs a sentence that ***the chosen action was executed*** at time ***t***

The Wumpus World Example

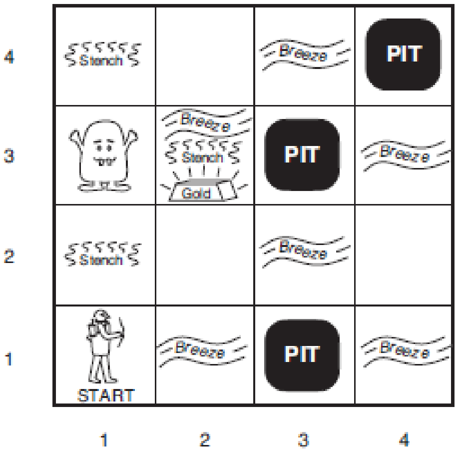


The Wumpus World Example





The Wumpus World



The PEAS description for Wumpus World:

Performance measure:

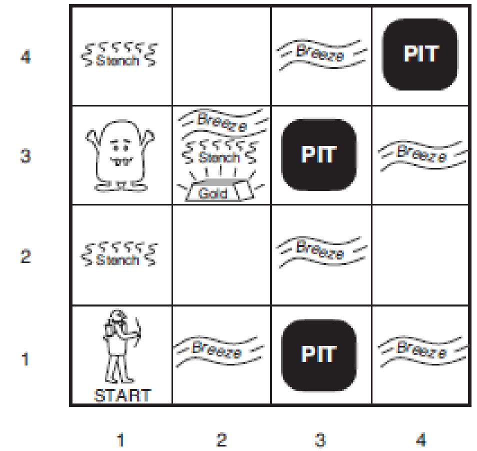
- +1000 for climbing out of the cave with the gold,
- –1000 for falling into a pit or being eaten by the Wumpus,
- –1 for each action taken
- –10 for using up the arrow

Environment:

- A 4×4 grid of rooms. The agent starts in the square labelled [1,1], facing to the right.

The game ends either when the agent dies or when the agent climbs out of the cave.

The Wumpus World



The PEAS description for Wumpus World:

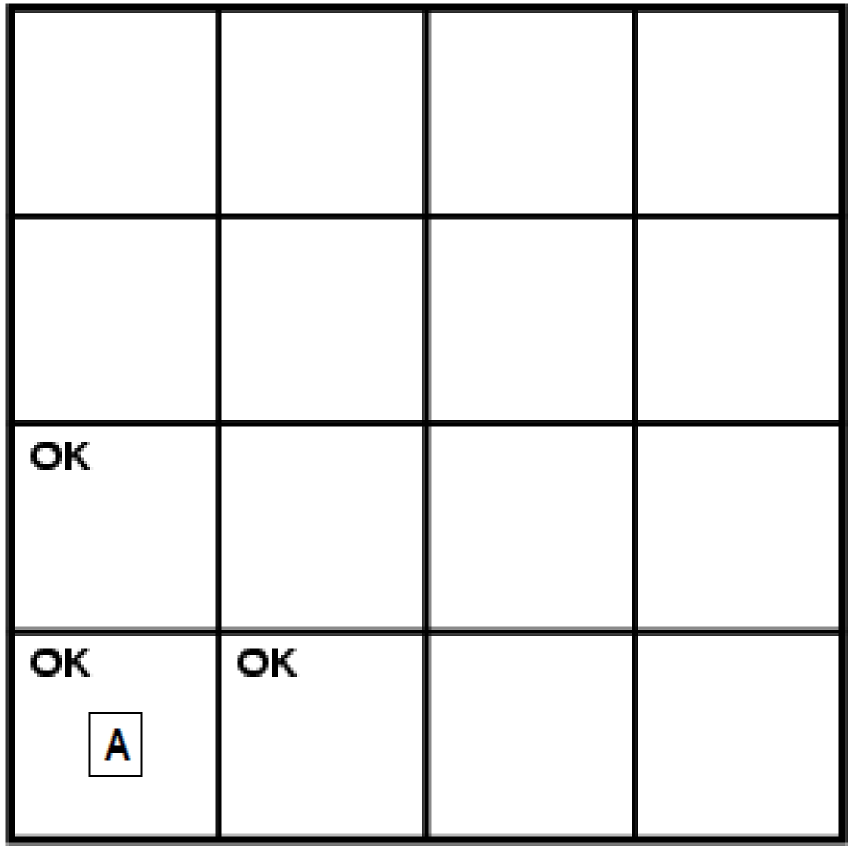
Actuators:

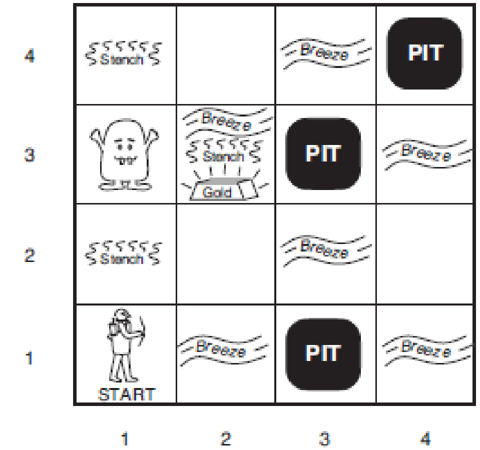
- The agent can move *Forward, TurnLeft by 90◦, TurnRight by 90◦*, grab, shoot

Sensors:

- The square adjacent directly (not diagonally) to the square containing **Wumpus**, the agent will perceive a ***Stench***.
- The squares adjacent to a **pit**, the agent will perceive a ***Breeze***.
- The square with **gold**, the agent will perceive a ***Glitter***.
- An agent **walks into a wall**, it will perceive a ***Bump***.
- When the **Wumpus is killed**, it emits a woeful ***Scream*.**

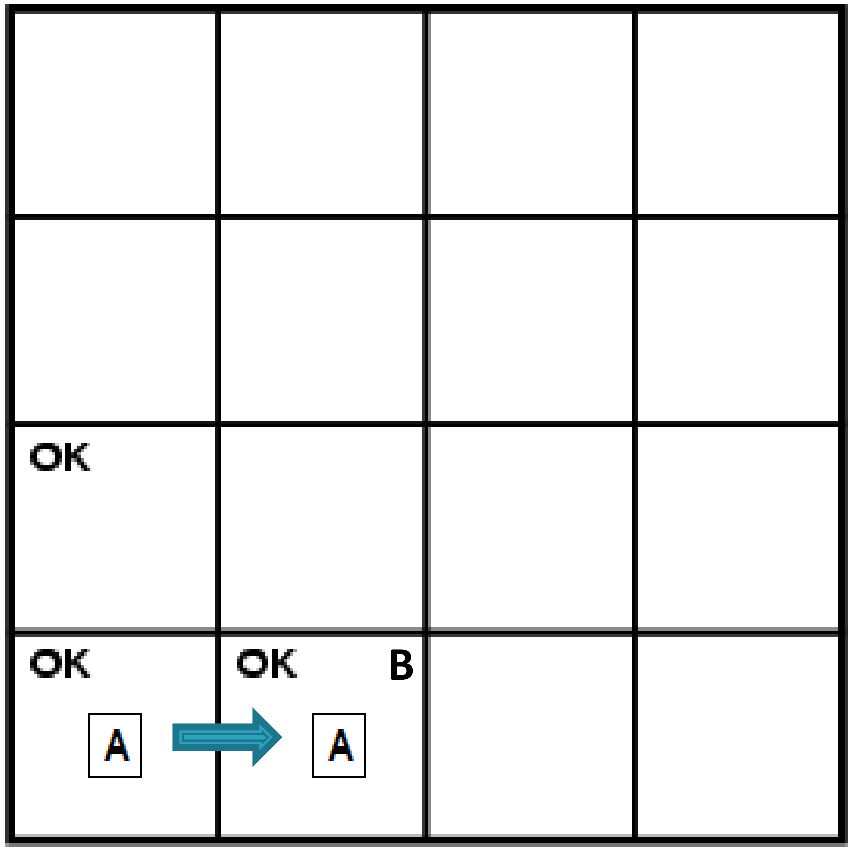
The Wumpus World

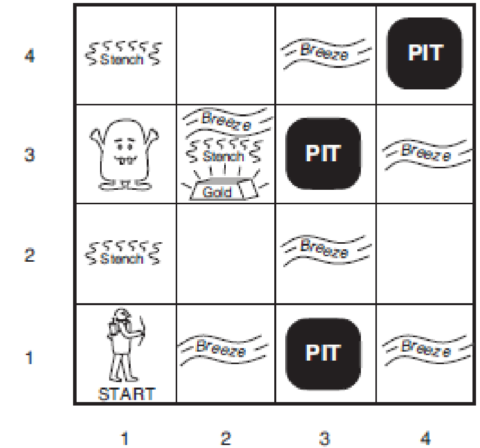




A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

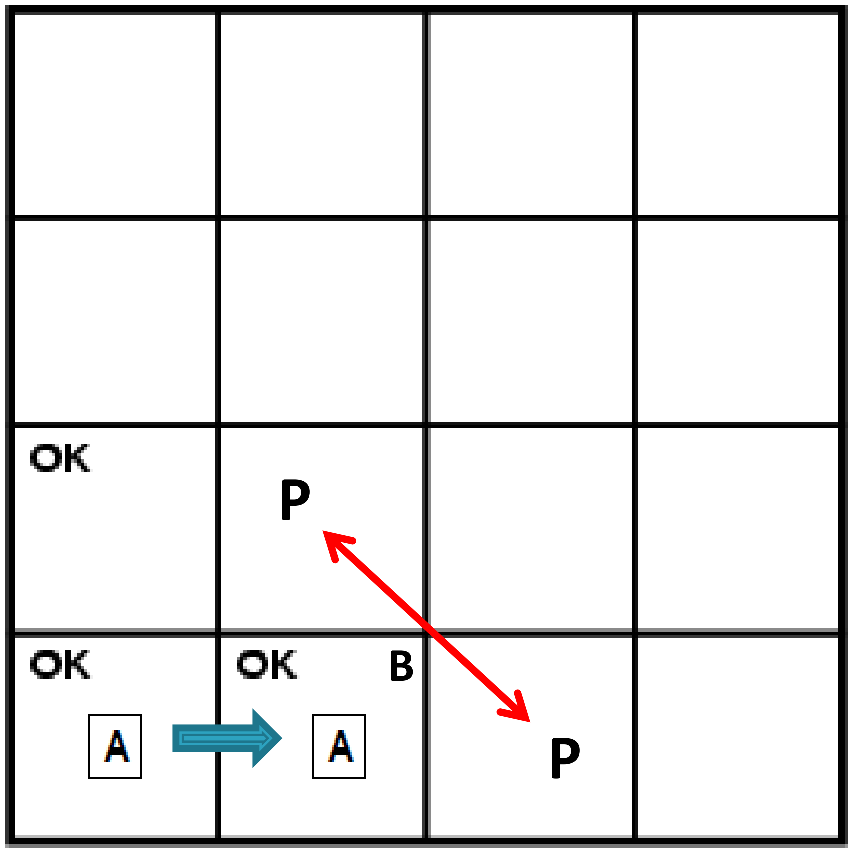
The Wumpus World

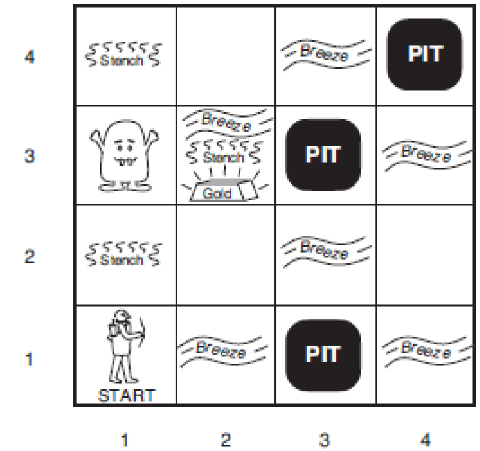




A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

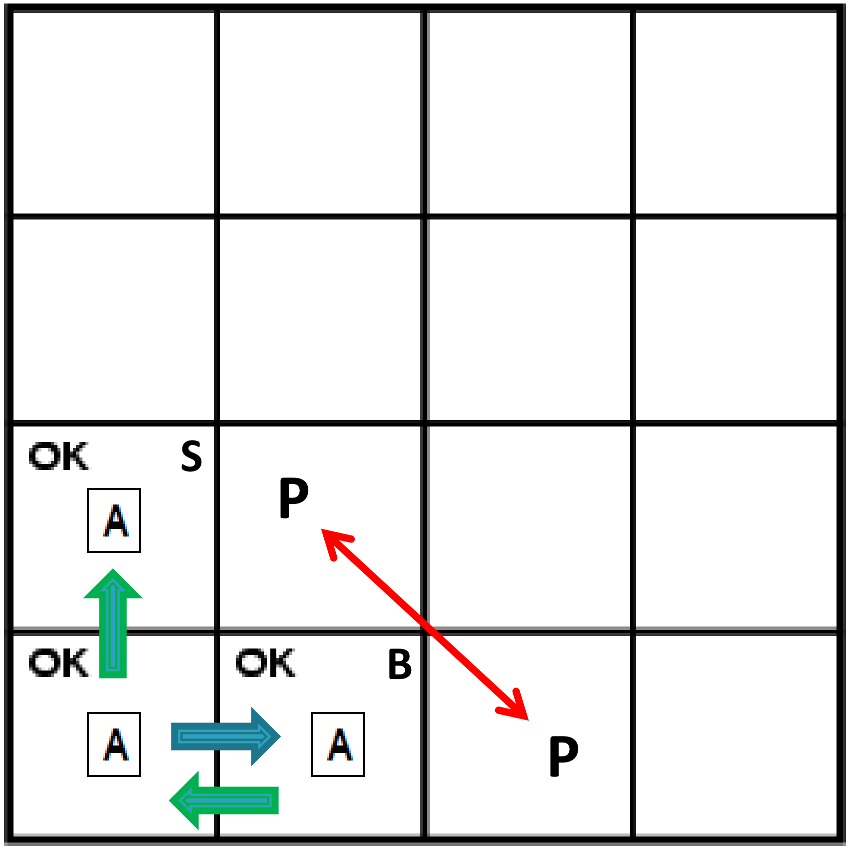
The Wumpus World

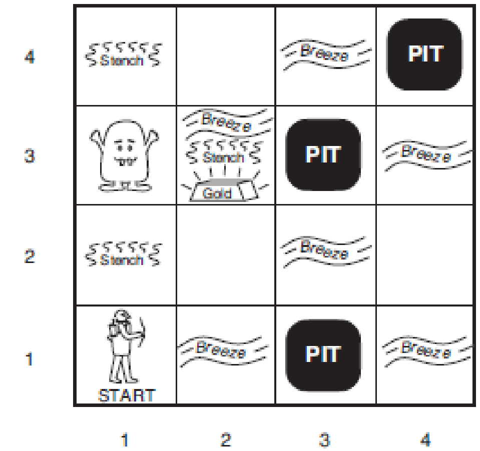




- **A** = *Agent*
- **B** = *Breeze*
- **G** = *Glitter, Gold*
- **OK** = *Safe square*
- **P** = *Pit*
- **S** = *Stench*
- **V** = *Visited*
- **W** = *Wumpus*

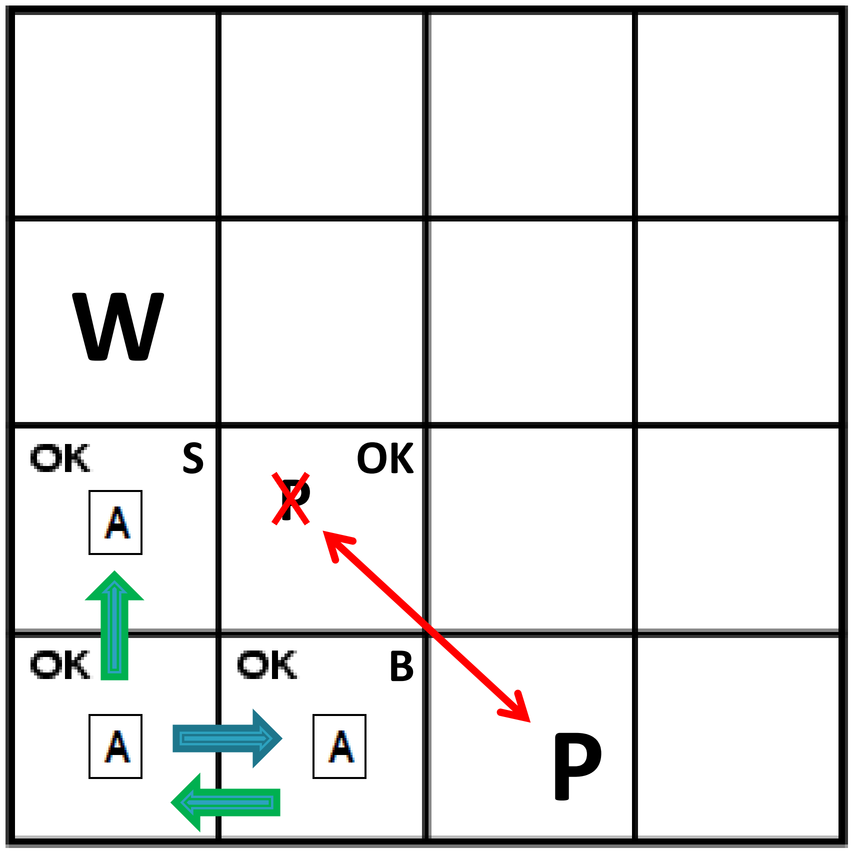
The Wumpus World

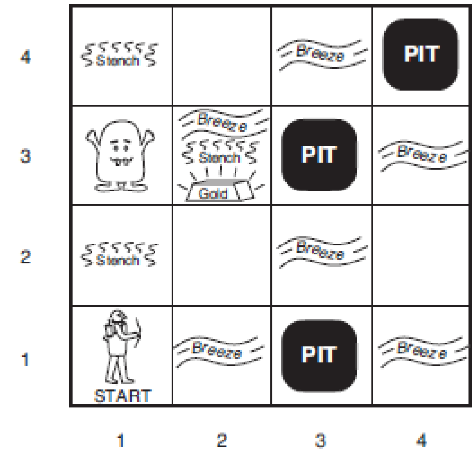




A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

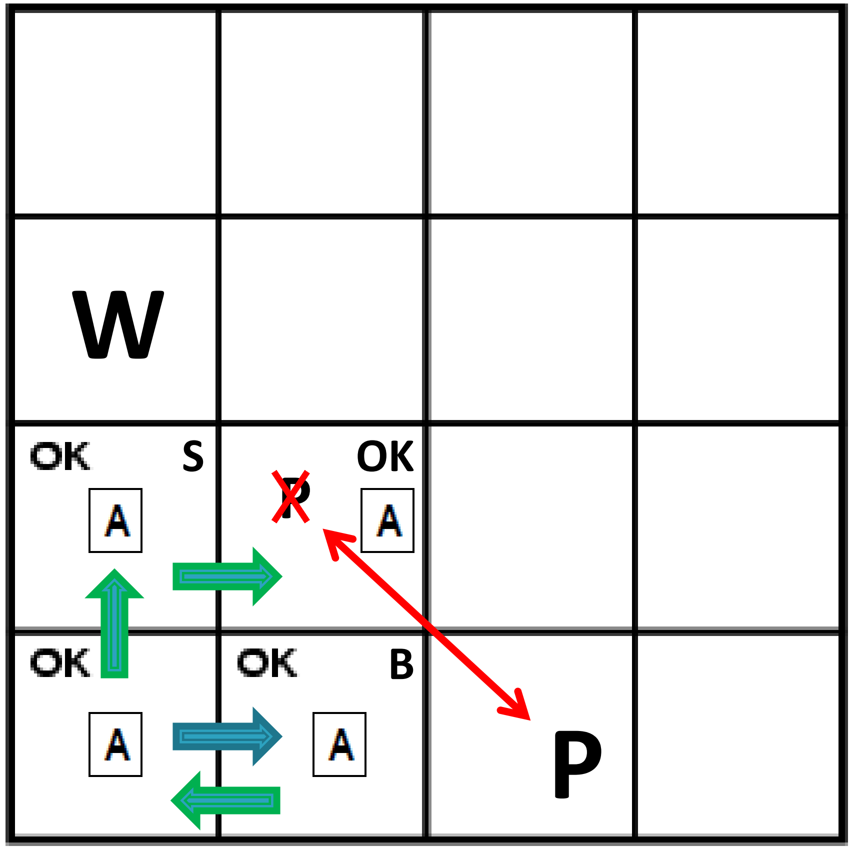
The Wumpus World

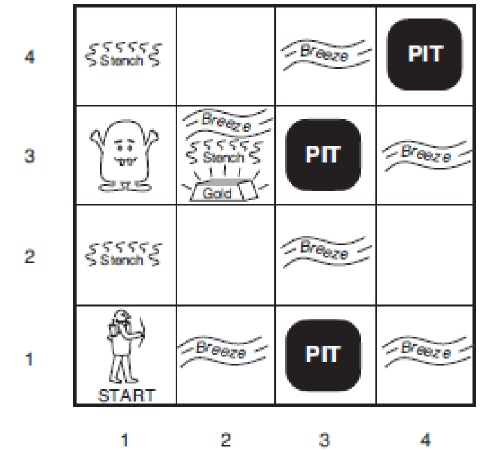




$\boxed{A}$ = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

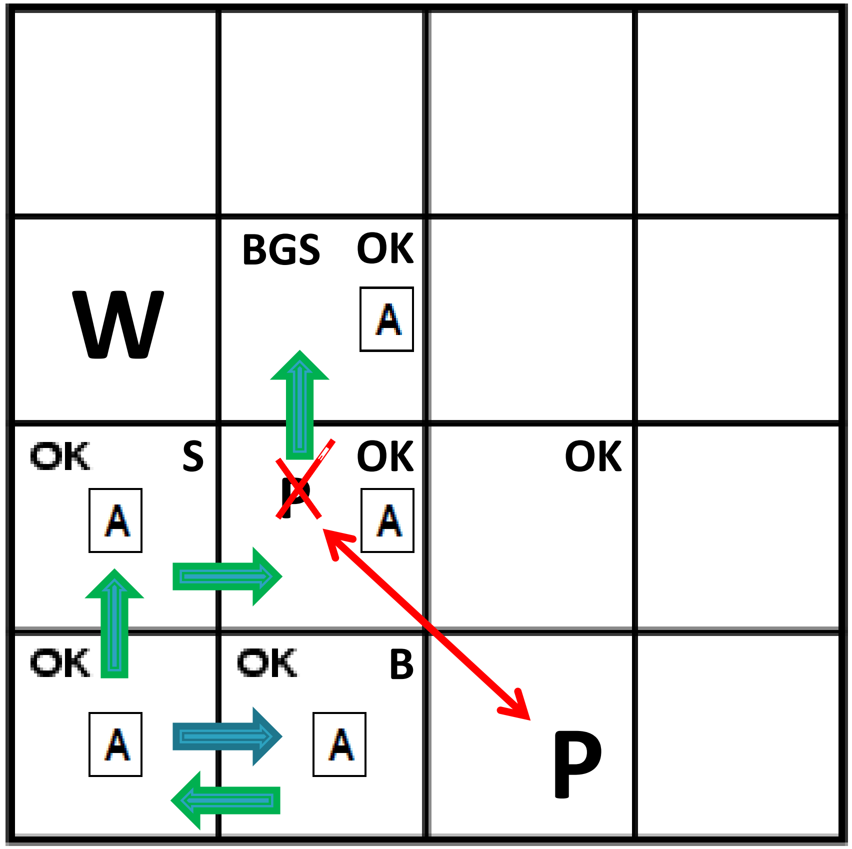
The Wumpus World

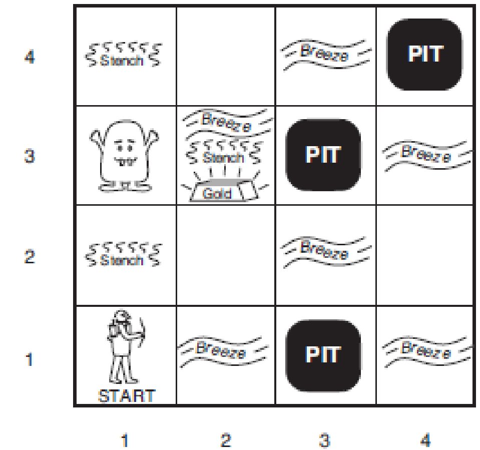




- $\boxed{A}$ = *Agent*
- **B** = *Breeze*
- **G** = *Glitter, Gold*
- **OK** = *Safe square*
- **P** = *Pit*
- **S** = *Stench*
- **V** = *Visited*
- **W** = *Wumpus*

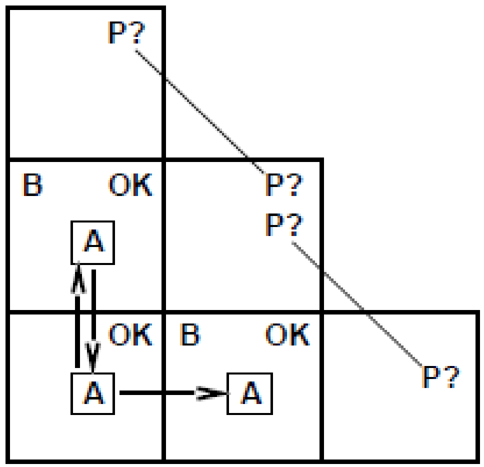
The Wumpus World





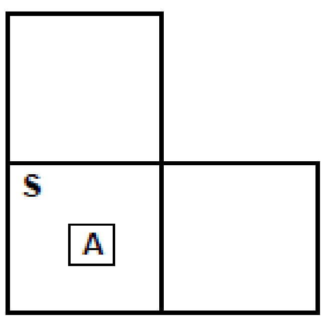
A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

The Wumpus World



Breeze in (1,2) and (2,1)
$\Rightarrow$ no safe actions

Assuming pits uniformly distributed,
(2,2) has pit w/ prob 0.86, vs. 0.31



Smell in (1,1)
$\Rightarrow$ cannot move
Can use a strategy of coercion:
shoot straight ahead
wumpus was there $\Rightarrow$ dead $\Rightarrow$ safe
wumpus wasn't there $\Rightarrow$ safe

Logic

Logic

- The knowledge bases consist of sentences.

How to represent these sentances?

- **Logic**, a formal language, is the solution --- a way of manipulating expressions in the language.

- Logic has
  - Syntax
  - Semantics

Logic

Syntax:

What expressions are legal --- what are allowed to write down.

- The notion of syntax is clear enough with the example: "$x + y = 4$" is a well-formed sentence, whereas "$x4y+=$" is not.

Semantics:

What legal expression means --- meaning of sentences

- the sentence "$x + y = 4$" is **true** in a **world** where x is 2 and y is 2, but **false** in a **world** where x is 1 and y is 1.
- *Syntax is a form and semantics is the content.*

Logic

Semantics:

- The semantics defines the **truth** of each sentence with respect to each **possible world**.
- The term **model** can be used in place of **"possible world."**
- If a sentence $\boldsymbol{\alpha}$ is true in model $\boldsymbol{m}$, we say that $\boldsymbol{m}$ **satisfies** $\boldsymbol{\alpha}$ or sometimes $\boldsymbol{m}$ is a **model** of $\boldsymbol{\alpha}$.
- The notation $\boldsymbol{M(\alpha)}$ --- the set of all **model**s of $\boldsymbol{\alpha}$.

Logic --- Entailment

Entailment:

- means that **one thing follows from another**:

$$\alpha \models \beta$$

- *if and only if, in every model in which* $\boldsymbol{\alpha}$ *is true,* $\boldsymbol{\beta}$ *is also true*. We can write

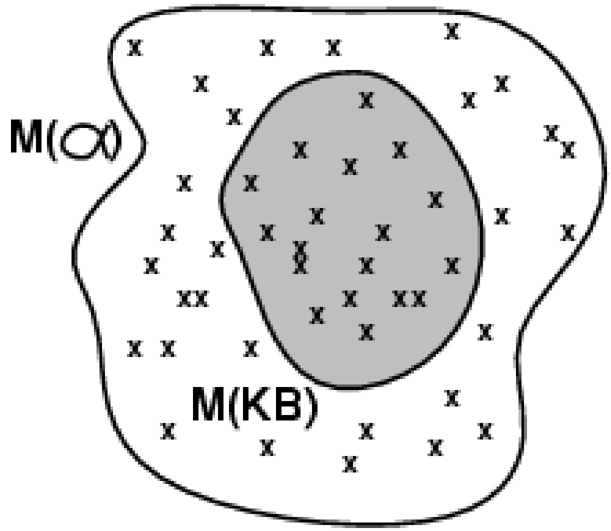
$$\boxed{\alpha \models \beta \text{ if and only if } M(\alpha) \subseteq M(\beta)}$$

- The notation $\subseteq$ means that: if $\boldsymbol{\alpha} \models \boldsymbol{\beta}$, then $\boldsymbol{\alpha}$ is a *stronger assertion than* $\boldsymbol{\beta}$

Logic --- Entailment

- We say m is a model of sentence α if α is true in m

- $M(\alpha)$ is the set of all models of α



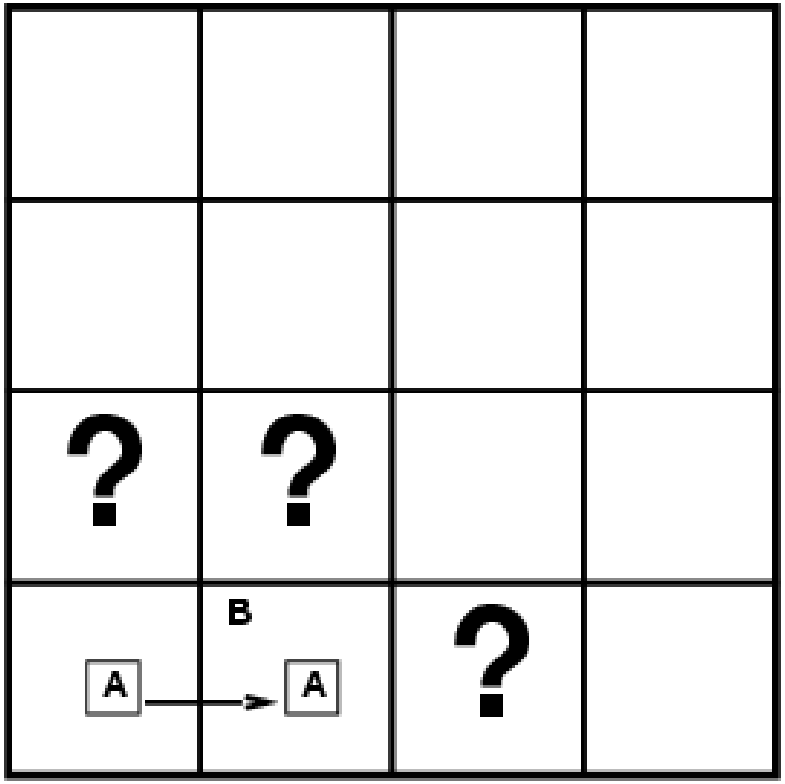
Then $KB \models \alpha$ iff $M(KB) \subseteq M(\alpha)$

Example:

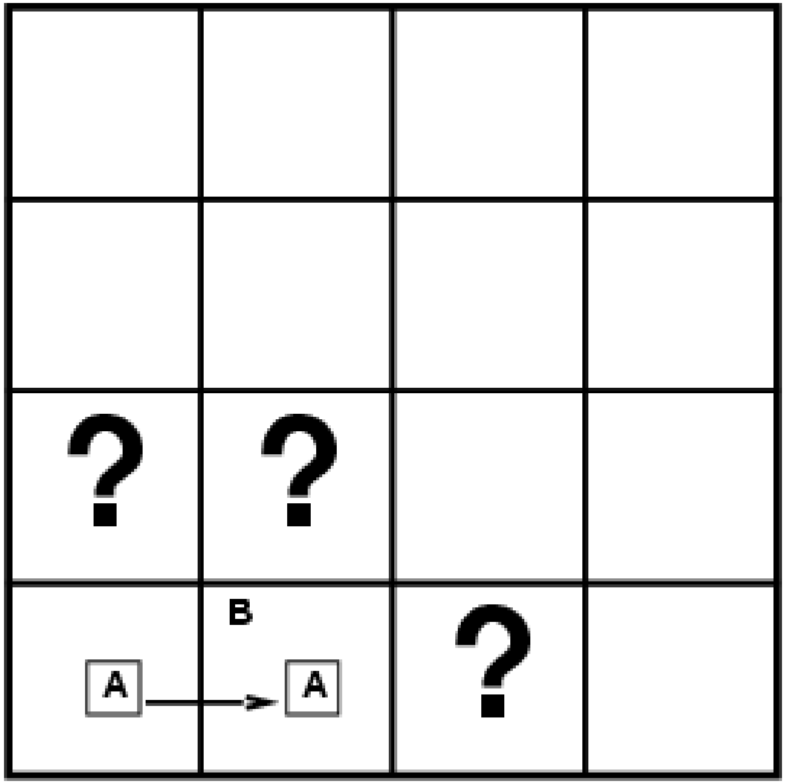
- The sentence $x = 0$ entails the sentence $xy = 0$
  - In any model where x is zero, it is the case that xy is zero (regardless of the value of y)

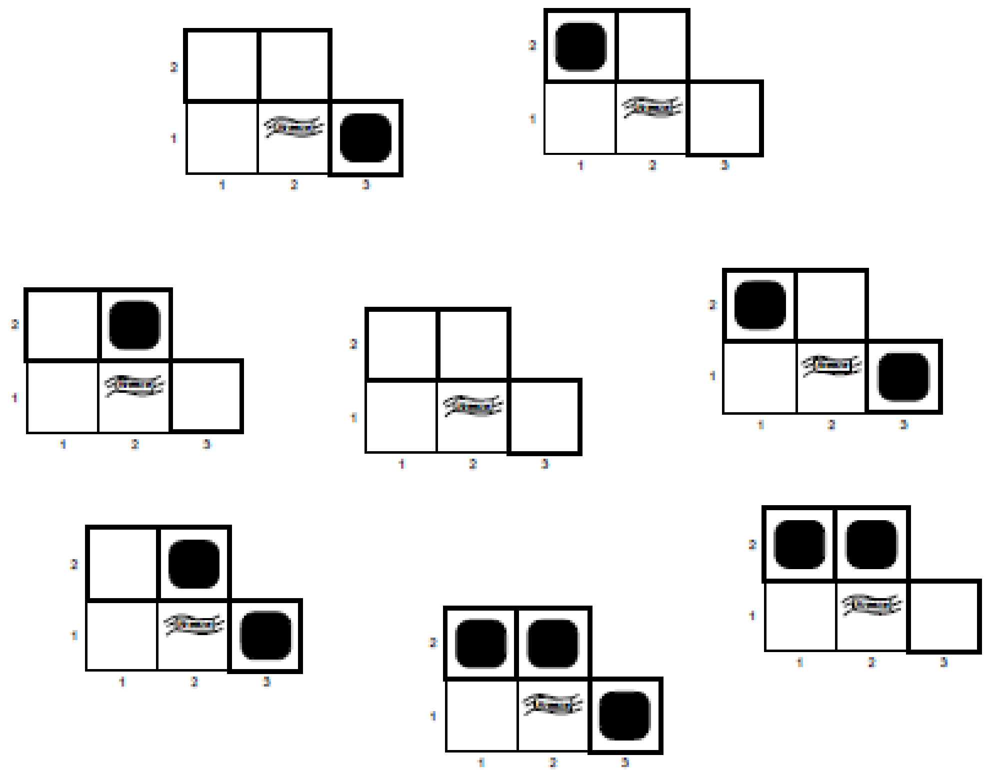
Entailment--- Wumpus World

- Situation after detecting nothing in [1,1], moving right, **breeze** in [1,2]
- Consider possible models for **KB** assuming only pits
- **3 Boolean choices** $\Rightarrow$ 8 possible models



Entailment--- Wumpus World

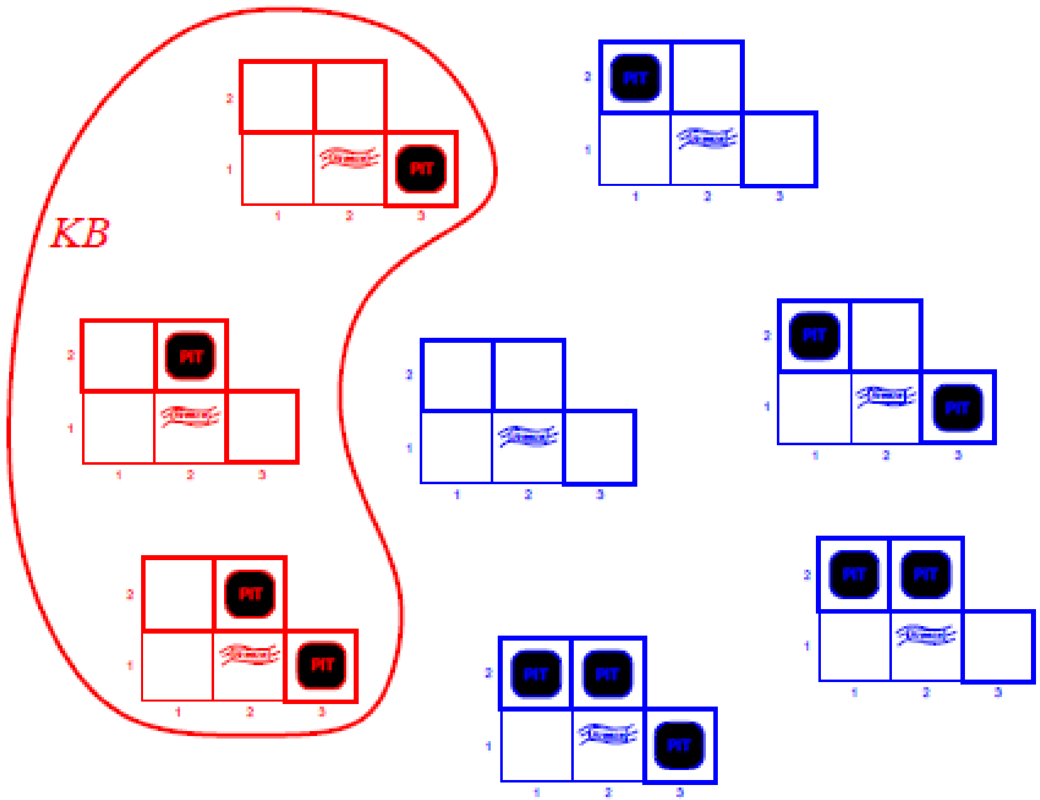




3 Boolean choices $\Rightarrow$ 8 possible models

regardless of wumpus-world rules

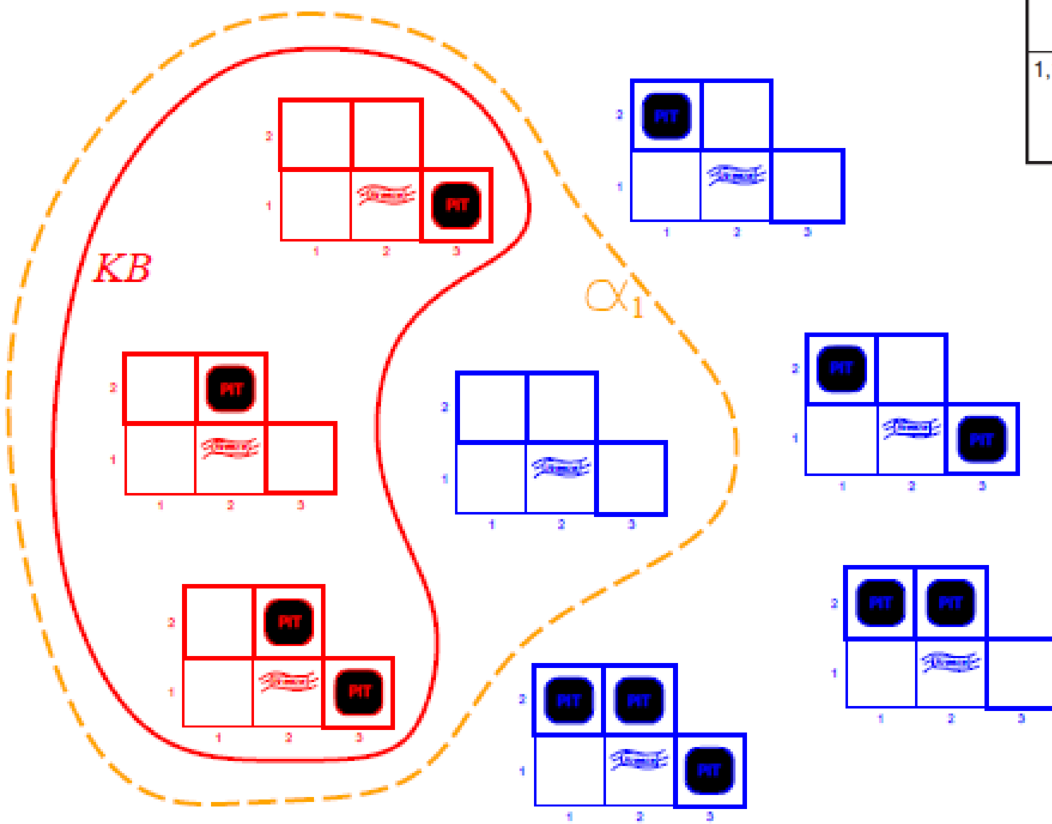
Entailment--- Wumpus World



KB = wumpus-world rules + observations

Entailment--- Wumpus World

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 $\boxed{A}$ OK	2,1 OK	3,1	4,1

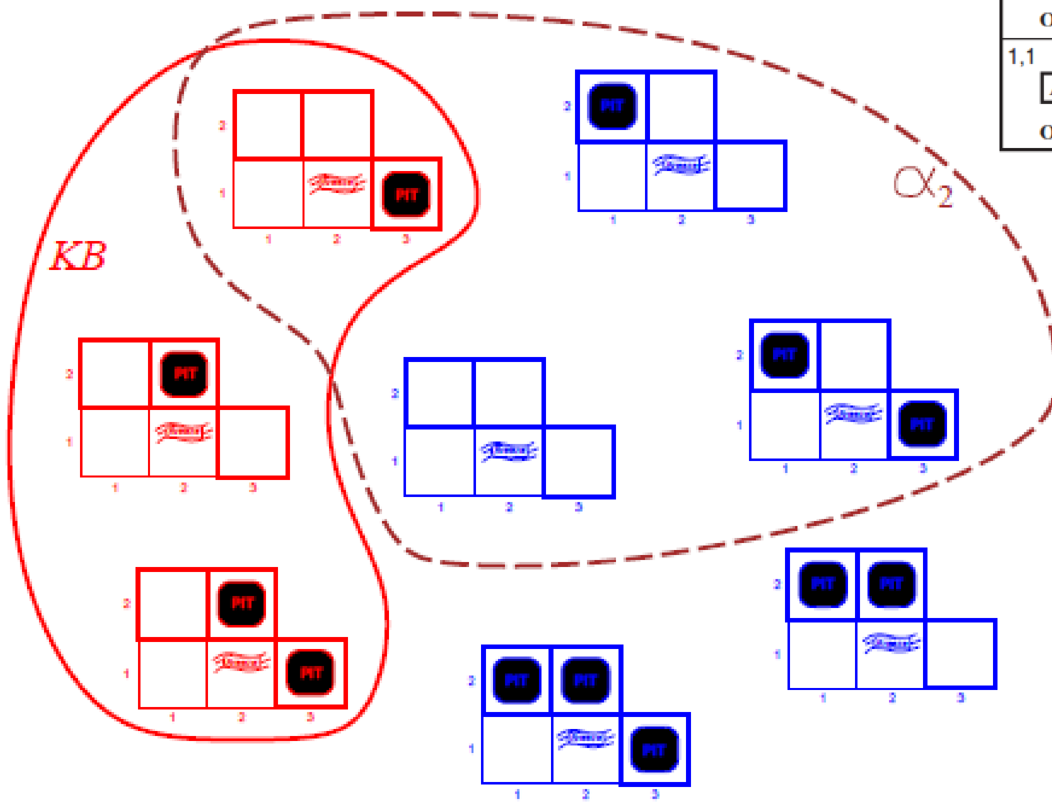


KB = wumpus-world rules + observations

α_1 = "[1,2] is safe", $KB \models \alpha_1$, proved by model checking

Entailment--- Wumpus World

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 $\boxed{A}$ OK	2,1 OK	3,1	4,1



KB = wumpus-world rules + observations

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

Inference

- *If an inference algorithm* $\boldsymbol{i}$ *can derive* $\boldsymbol{\alpha}$ *from* $\boldsymbol{KB}$, we write

$$KB \vdash_i \alpha$$

- which is pronounced "**$\boldsymbol{\alpha}$ is derived from $\boldsymbol{KB}$ by $\boldsymbol{i}$**" or "**$\boldsymbol{i}$ derives $\boldsymbol{\alpha}$ from $\boldsymbol{KB}$**."

Soundness:

- An inference algorithm that **derives only entailed sentences** is called **sound or truth preserving.**
- Soundness is a highly desirable property.

Completeness:

- An inference algorithm is complete if it can derive any sentence that is **entailed**.

Logic

- **We'll look at two kinds of logic:**

Propositional Logic

which is relatively simple.

First-order Logic

which is more complicated.

Propositional Logics

Propositional Logic: Syntax

- The **syntax** of propositional logic defines the *allowable sentences*.

What are the sentances?

- Sentance are well formed formulas
- **True** and **False** are sentances
- Propositional variables are sentences. P, Q, R, S etc.

Propositional Logic: Syntax

- The **atomic sentences** consist of a **single proposition symbol**.
- Each such symbol stands for a proposition that can be **True or False**.
- The **complex sentences** are constructed from simpler sentences, using parentheses and **logical connectives**.
- There are five connectives in common use:

  $\neg$ (not), $\wedge$ (and), $\vee$ (or),

  $\Rightarrow$ (implies), $\Leftrightarrow$ (if and only if)

Propositional Logic: Syntax

- **¬ (not)** A sentence such as $\neg W_{1,3}$ is called the negation of $W_{1,3}$.
  - A **literal** is either an atomic sentence (a positive literal) or a negated atomic sentence (a negative literal).
- **∧ (and)** A sentence whose main connective is ∧, such as $W_{1,3} \wedge P_{3,1}$, is called a conjunction; its parts are the *conjuncts*.
- **∨ (or)** A sentence using ∨, such as $(W_{1,3} \wedge P_{3,1}) \vee W_{2,2}$, is a disjunction of the *disjuncts* $(W_{1,3} \wedge P_{3,1})$ and $W_{2,2}$.

Propositional Logic: Syntax

- **$\Rightarrow$ (implies)** A sentence such as **$(W_{1,3} \wedge P_{3,1}) \Rightarrow \neg W_{2,2}$** is called an implication (or conditional). The premise or antecedent is **$(W_{1,3} \wedge P_{3,1})$**.
- Implications are also known as rules or **if–then** statements.
- The implication symbol is sometimes written as $\supset$ or $\rightarrow$ or $\Rightarrow$.
- **$\Leftrightarrow$ (if and only if)** The sentence **$W_{1,3} \Leftrightarrow \neg W_{2,2}$** is a bi-conditional. Sometime it is written as $\equiv$.

Propositional Logic: Syntax

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)

BNF (Backus–Naur Form) Grammar

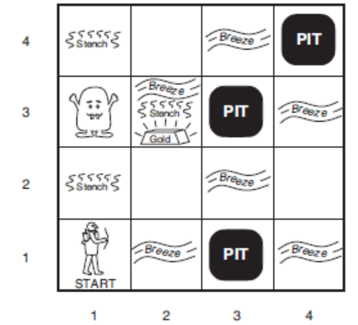
$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & True \mid False \mid P \mid Q \mid R \mid \ldots \\
ComplexSentence & \rightarrow & (\ Sentence\ ) \mid [\ Sentence\ ] \\
 & \mid & \neg\ Sentence \\
 & \mid & Sentence \land Sentence \\
 & \mid & Sentence \lor Sentence \\
 & \mid & Sentence \ \Rightarrow\ Sentence \\
 & \mid & Sentence \ \Leftrightarrow\ Sentence \\
 & & \\
\text{OPERATOR PRECEDENCE} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

BNF (Backus–Naur Form) Grammar

Precedence Example:

$A \lor B \land C$	$A \lor (B \land C)$
$A \land B \to C \lor D$	$(A \land B) \to (C \lor D)$
$A \to B \lor C \leftrightarrow D$	$(A \to (B \lor C)) \leftrightarrow D$

Propositional Logic: Semantics



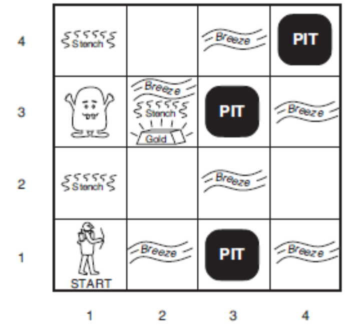
- **The semantics** defines **the rules** for determining the truth value of a sentence with respect to a particular model.
- In propositional logic, a model simply fixes the truth value—**true** or **false**—for every proposition symbol

For example:

- If the sentences in the knowledge base make use of the proposition symbols $P_{1,2}$, $P_{2,2}$, and $P_{3,1}$, then one possible model is:

$$m_1 = \{P_{1,2} = false,\ P_{2,2} = false,\ P_{3,1} = true\}$$

Propositional Logic: Semantics



- The semantics for propositional logic must specify **how to compute the truth value** of *any sentence, given a model.*

For Atomic sentences:

- **True** is true in every model and **False** is false in every model.
- *The truth value of every other proposition symbol must be specified directly in the model.*
  - For example, in the model $\boldsymbol{m_1}$ given earlier, $\boldsymbol{P}_{1,2}$ is false.

$$m_1 = \{P_{1,2} = false,\ P_{2,2} = false,\ P_{3,1} = true\}$$

Propositional Logic: Semantics

For complex sentences

- We have five rules, which hold for any sub-sentences $\boldsymbol{P}$ and $\boldsymbol{Q}$ in any model $\boldsymbol{m}$

- $\neg P$ is true iff P is false in m.
- $P \wedge Q$ is true iff both P and Q are true in m.
- $P \vee Q$ is true iff either P or Q is true in m.
- $P \Rightarrow Q$ is false unless P is true and Q is false in m.
- $P \Leftrightarrow Q$ is true iff P and Q are both true or both false in m.

Propositional Logic: Semantics

- The propositional logic **does not require any relation of *causation* or *relevance*** between P and Q.
  - For example, the sentence **"5 is odd implies Tokyo is the capital of Japan"** is a true sentence of propositional logic, even though it is not a well-formed English sentence.

- In case of implication, **any implication is true whenever its antecedent is false**.
  - For example, **"5 is even implies Sam is smart"** is true, regardless of whether Sam is smart or not.

Propositional Logic: Truth Table

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$Q \rightarrow P$	$P \leftrightarrow Q$
f	f	t	f	f	t	t	t
f	t	t	f	t	t	f	f
t	f	f	f	t	f	t	f
t	t	f	t	t	t	t	t

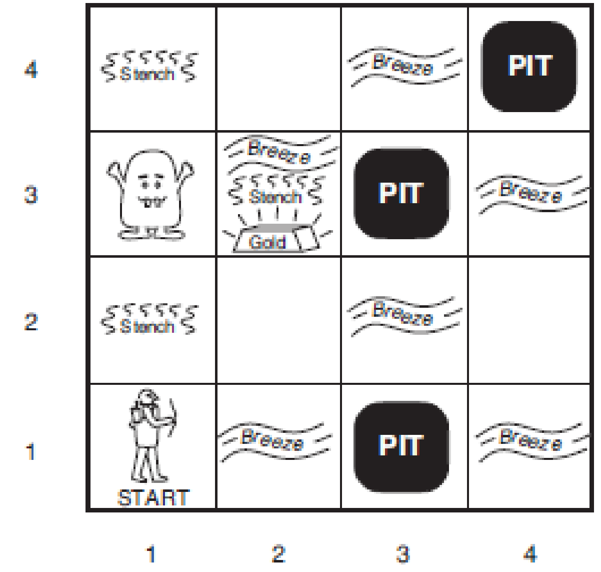
A simple knowledge base

- With propositional logic, we can construct a knowledge base for the Wumpus world.

For Example:

$P_{x,y}$ 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.
$B_{x,y}$ is true if the agent perceives a breeze in $[x, y]$.
$S_{x,y}$ is true if the agent perceives a stench in $[x, y]$.

A simple knowledge base



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

$$R_1 : \quad \neg P_{1,1} \,.$$

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

$$R_2 : \quad B_{1,1} \quad \Leftrightarrow \quad (P_{1,2} \lor P_{2,1}) \,.$$
$$R_3 : \quad B_{2,1} \quad \Leftrightarrow \quad (P_{1,1} \lor P_{2,2} \lor P_{3,1}) \,.$$

Standard Logical Equivalences

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{De Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{De Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Reading Material

- **Artificial Intelligence,** A Modern Approach

  Stuart J. Russell and Peter Norvig

  - **Chapter 7.**