

# **CSCI 561**

## **Foundation for Artificial Intelligence**

### **9-10: Knowledge, Reasoning, Propositional Logic**

#### **Discussion: Review for Midterm1**

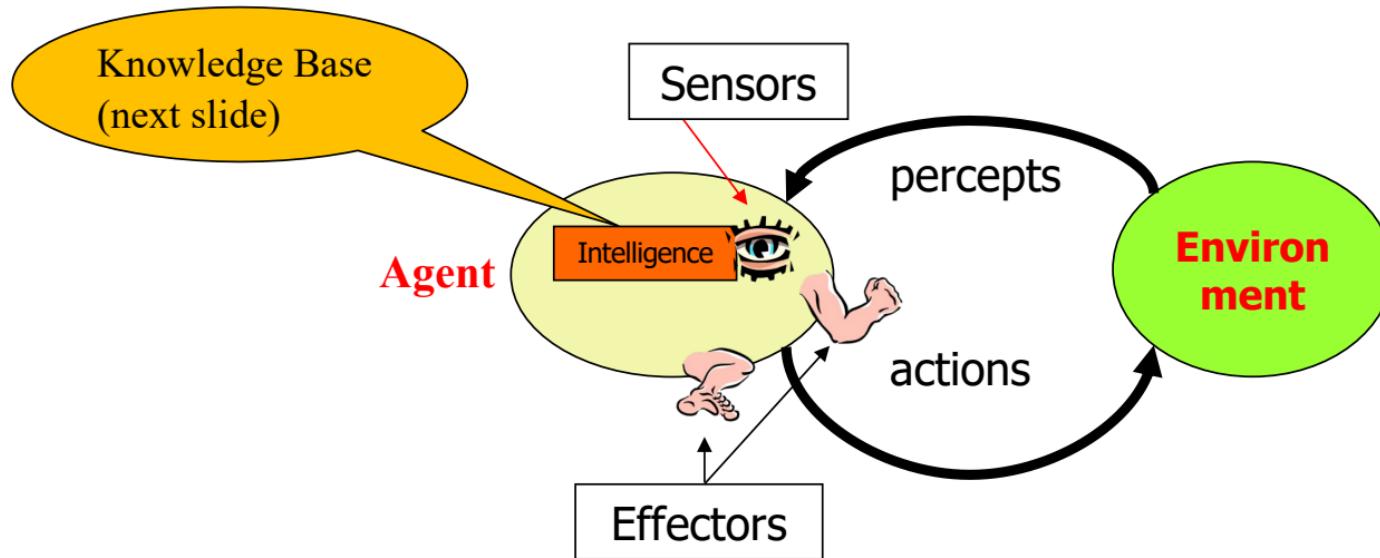
#### **Knowledge and Reasoning**

- Knowledge representation
- Logic and representation
- Propositional (Boolean) logic
- Normal forms
- Inference in propositional logic
- Wumpus world example

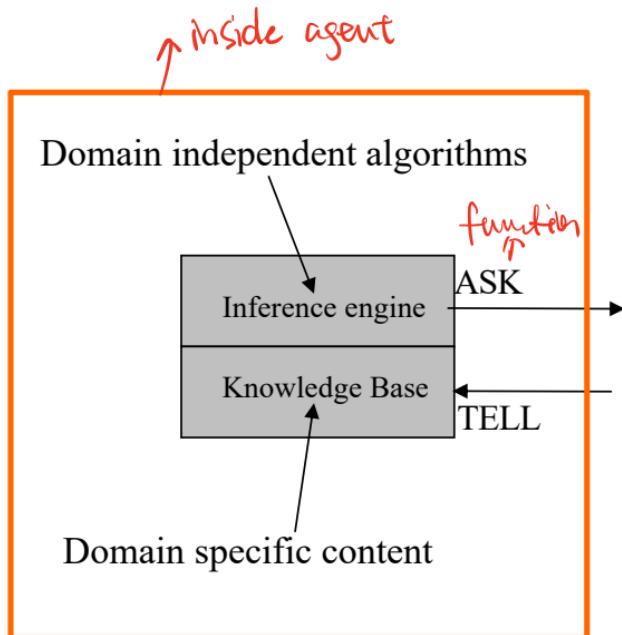
-> knowledge representation

1. Why

# Agent and Environment



# Knowledge-Based Agent



- Agent that uses **prior** or **acquired** knowledge to achieve its goals
  - Can make more efficient decisions
  - Can make informed decisions
- **Knowledge Base (KB):** contains a set of representations of facts about the Agent's environment
- Each representation is called a **sentence**
- Use some **knowledge representation language (KRL)**, to TELL it what to know, and what to do, e.g., (temperature 72F) <sup>agent</sup>
- ASK agent to query what to do
- Agent can use inference to deduce new facts from the TELLED facts

# Generic Knowledge-Based 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)) tell agent what you perceived
    action  $\leftarrow$  ASK(KB, MAKE-ACTION-QUERY(t)) ask agent what you like to do
    TELL(KB, MAKE-ACTION-SENTENCE(action, t)) tell agent to do ask returned
    t  $\leftarrow$  t + 1
    return action
```

1. TELL KB what was perceived  
Uses a KRL to insert new sentences, representations of facts, into KB
2. ASK KB what to do  
Uses logical reasoning to examine actions and select the best action

2.leg:

## Wumpus World Example

Percepts Breeze, Glitter, Smell

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

Goals Get gold back to start  
without entering pit or wumpus square

4	Smell		Breeze	PIT
3	Wumpus	Breeze Smell	PIT	Breeze
2	Smell		Breeze	
1	START	Breeze	PIT	Breeze
	1	2	3	4



### Environment

Squares adjacent to wumpus are smelly

Squares adjacent to pit are breezy

Glitter if and only if gold is in the same square

Shooting kills the wumpus if you are facing it

Shooting uses up the only arrow

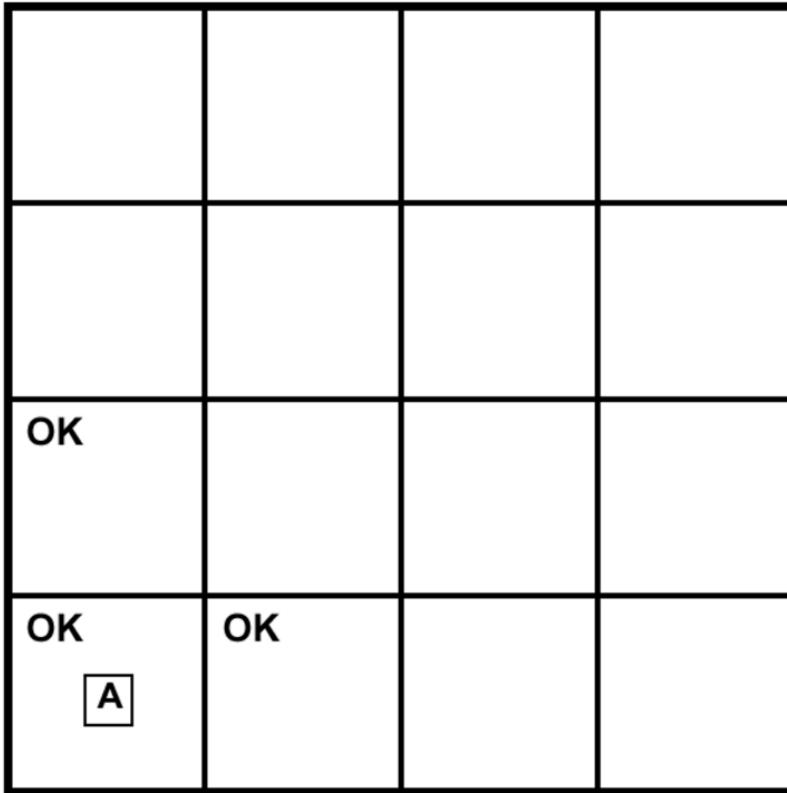
Grabbing picks up the gold if in the same square

Releasing drops the gold in the same square

## Wumpus World Characterization

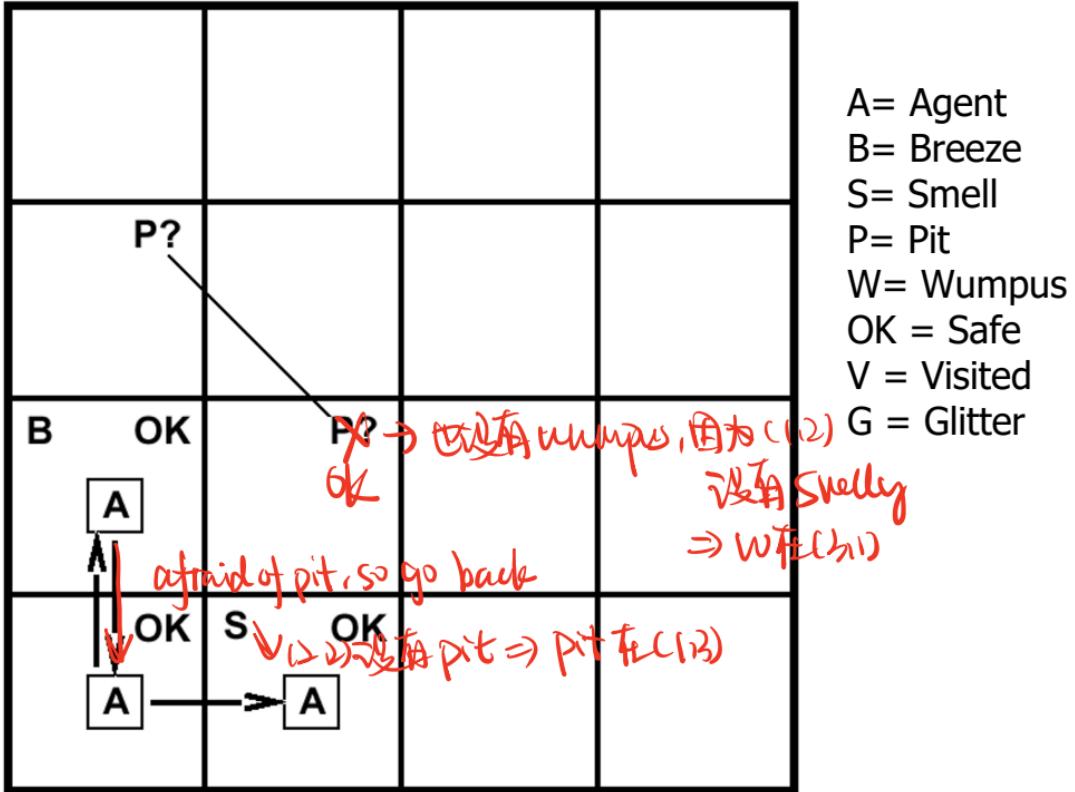
- Deterministic? Yes – outcome exactly specified
- Accessible? No – only local perception
- Static? Yes – Wumpus and pits do not move
- Discrete? Yes
- Episodic? (No) – because static

# Exploring a Wumpus World

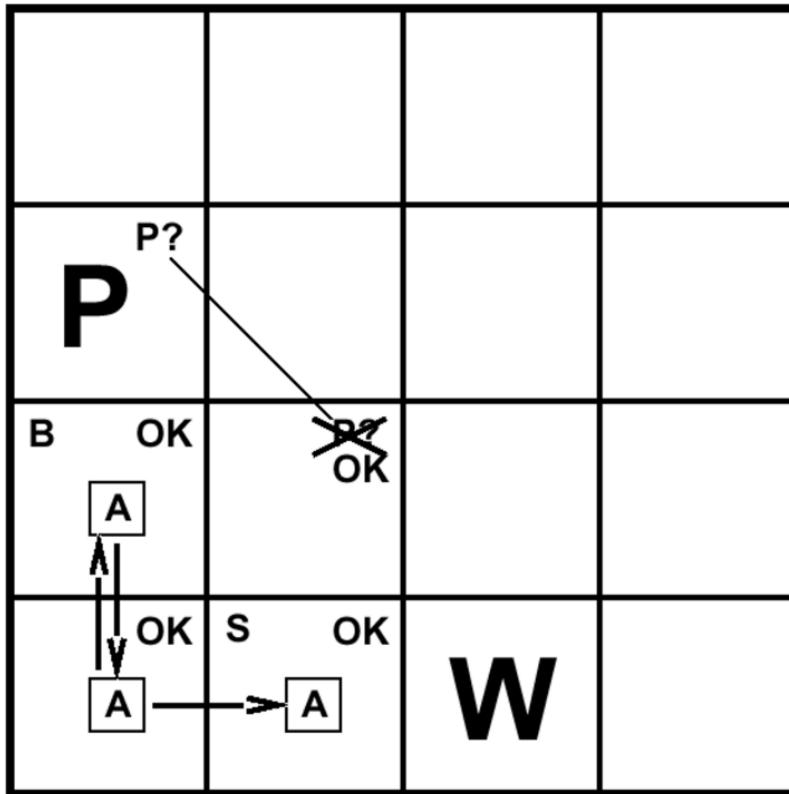


A= Agent  
B= Breeze  
S= Smell  
P= Pit  
W= Wumpus  
OK = Safe  
V = Visited  
G = Glitter

# Exploring a Wumpus World

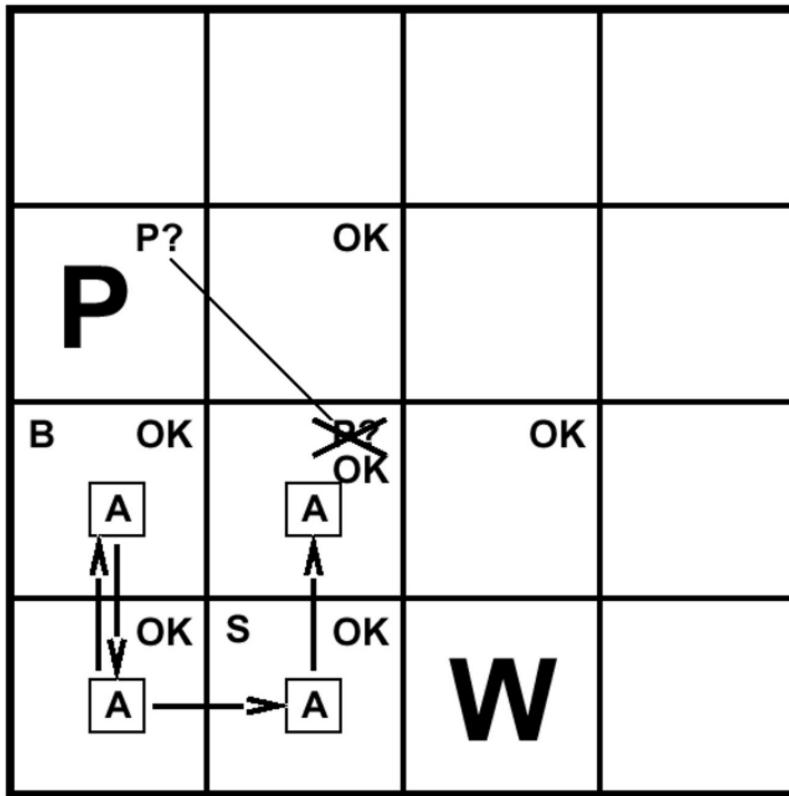


# Exploring a Wumpus World

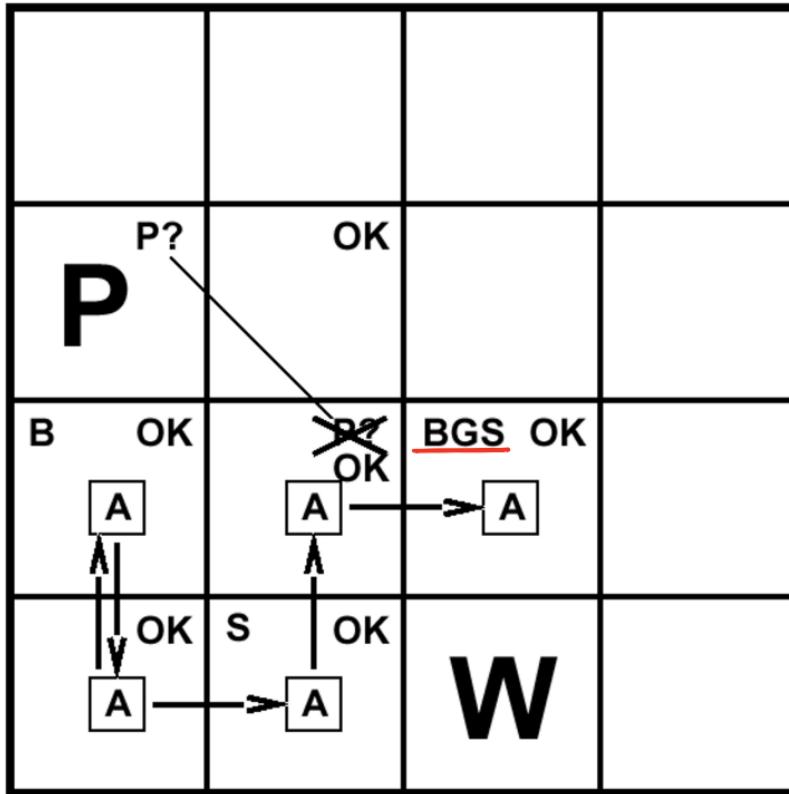


A= Agent  
B= Breeze  
S= Smell  
P= Pit  
W= Wumpus  
OK = Safe  
V = Visited  
G = Glitter

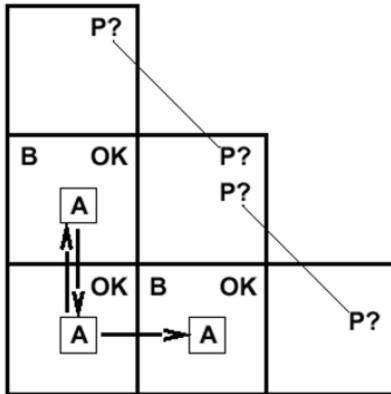
# Exploring a Wumpus World



# Exploring a Wumpus World

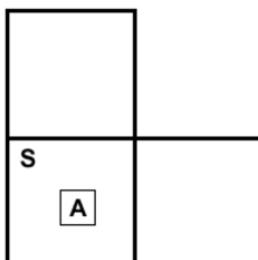


## Other Tight Spots



Breeze in (1,2) and (2,1)  
⇒ no safe actions

Assuming pits uniformly distributed,  
(2,2) is most likely to have a pit



Smell in (1,1)  
⇒ cannot move

Can use a strategy of coercion:  
shoot straight ahead  
wumpus was there ⇒ dead ⇒ safe  
wumpus wasn't there ⇒ safe

## Another Example Solution

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1 A OK	2,1 OK	3,1	4,1

No perception  1,2 and 2,1 OK

Move to 2,1

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

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
OK			
1,1 V OK	2,1 B OK	3,1 P?	4,1

B in 2,1  2,2 or 3,1 P?

1,1 V  no P in 1,1

Move to 1,2 (only option)

# Example Solution

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P! V OK	4,1

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

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P! V OK	4,1

S and No S when in 2,1  $\square$  1,3 or 1,2 has W

1,2 OK  $\square$  1,3 W

No B in 1,2  $\square$  2,2 OK & 3,1 P

## 2. Logic

### Logic in General

1. (1) Logics are formal languages for representing information such that conclusions can be drawn

Syntax defines the sentences in the language

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

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  $x = 0, y = 6$

## (2) Types of Logic

Logics are characterized by what they commit to as "primitives"

Ontological commitment: what exists—facts? objects? time? beliefs?

Epistemological commitment: what states of knowledge?

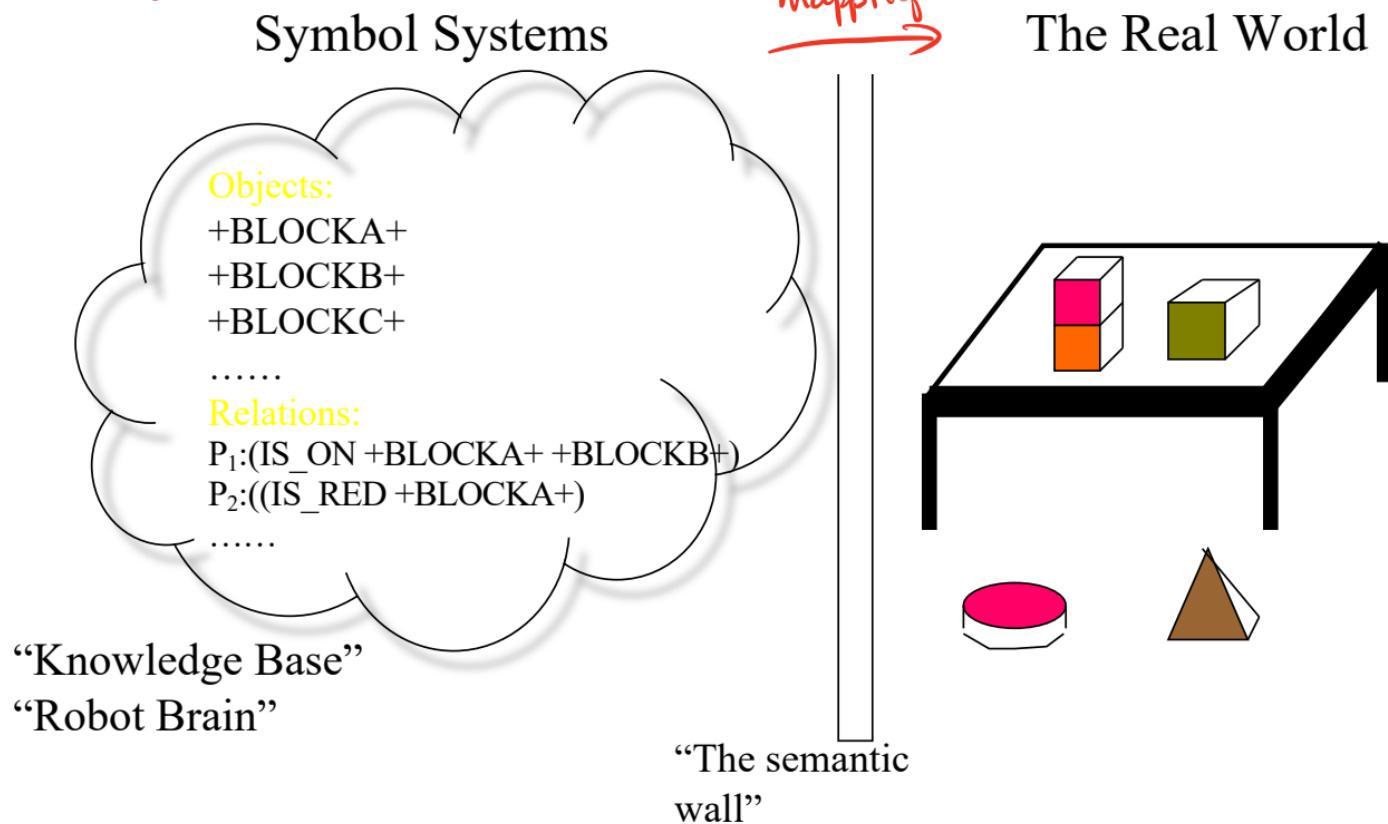
Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0...1
Fuzzy logic	degree of truth	degree of belief 0...1

### ③ Key Concepts of Logic

- Syntax
- Semantics
- Entailment
- Inference
- Soundness
- Completeness
- Inference Rules
- Normal Forms
- Truth Tables
- Reasoning

# Logic Representation of the Real World

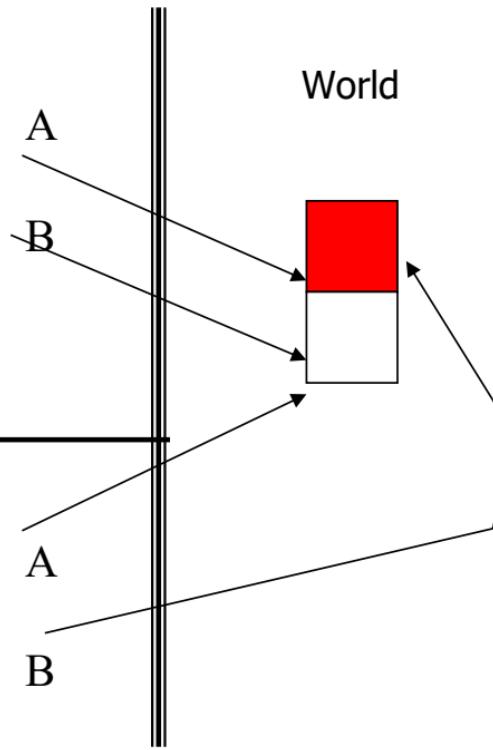
In agent brain



# Truth Depends on Interpretation

Representation 1

$ON(A,B)$  T  
 $ON(B,A)$  F



$ON(A,B)$  F

$ON(B,A)$  T

# The Key Logic Concepts

## *Entailment* $\models$

## about meaning

## Inference $\vdash$

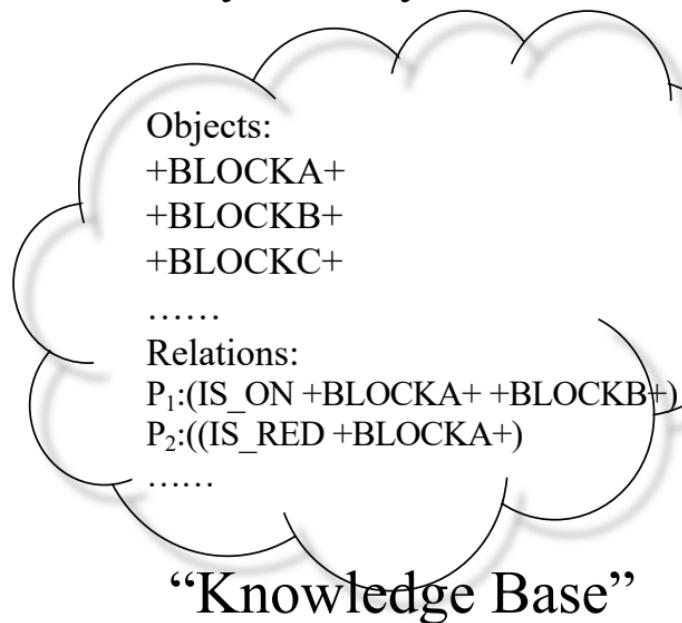
## about reason

4	 Stench		 Breeze	PIT
3		 Breeze  Stench  Gold		 Breeze
2	 Stench		 Breeze	
1	 START	 Breeze		 Breeze

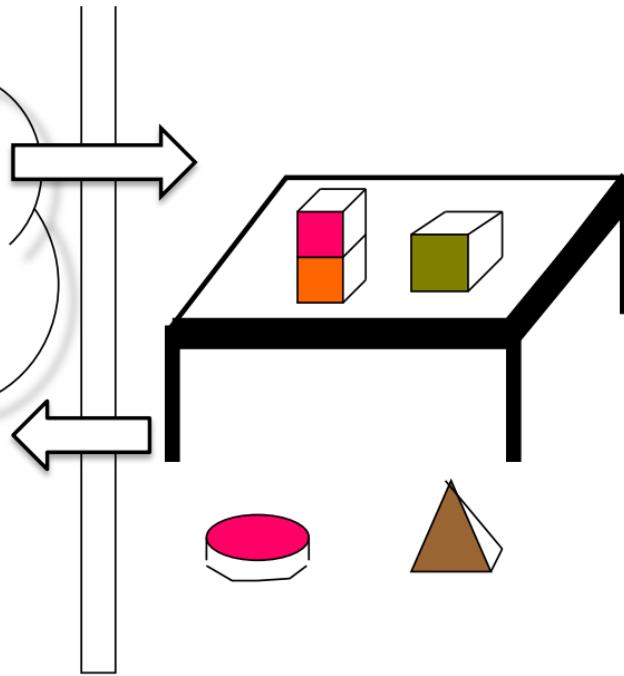
# Break the Semantic Wall

*mapping to real world*

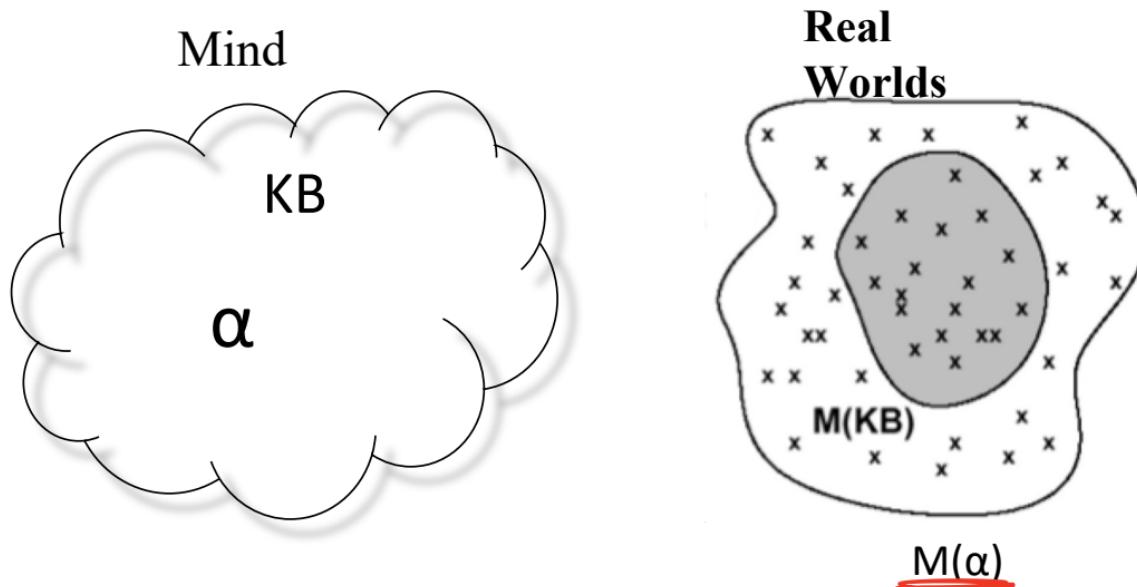
Symbol Systems



The Real World



# KB (“mind”) and Models (“real worlds”)



2.

## Entailment

只要 KB 真,  $\alpha$  真

(1)

$$KB \models \alpha$$

Knowledge base  $KB$  entails sentence  $\alpha$   
if and only if

$\alpha$  is true in all worlds where  $KB$  is true

E.g., the KB containing “the Giants won” and “the Reds won”  
entails “Either the Giants won or the Reds won”

## Entailment: How to Check?

Tippin

- $|KB| = \alpha ?$

- Is  $M(KB)$  a subset of  $M(\alpha)$  ?

go to  
real world

Inference 不需要,

just apply rules in head

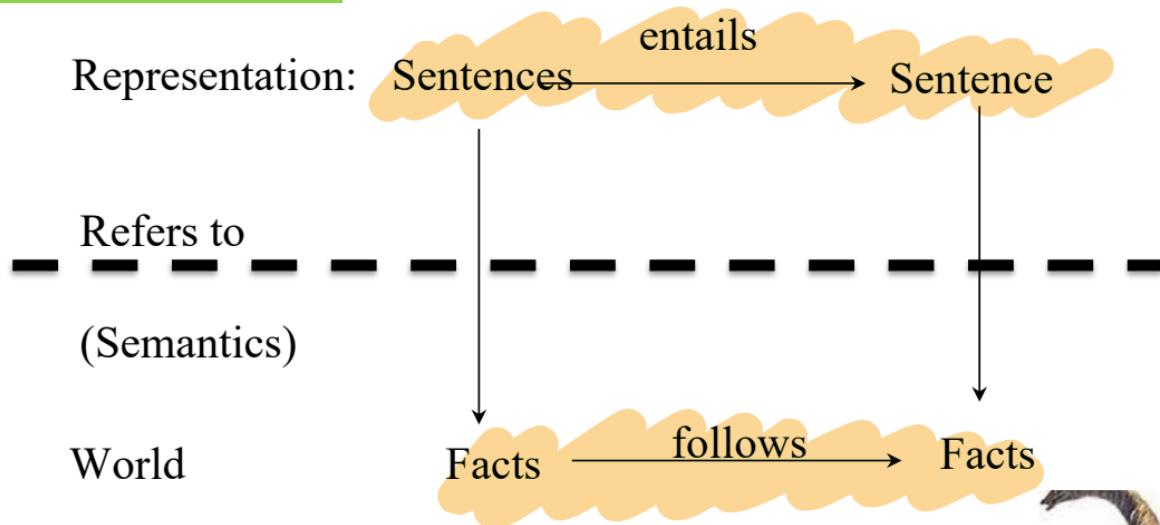
“Entailment” is different from “Inference”

Entailment means it is impossible for the following case to occur:  
**the premise ( $KB$ ) is true but the consequence ( $\alpha$ ) is false**

# Logic as a Representation of a World

e.g., “John has three brains.”

“how many brains did Argentinosaurus had?”



Really? Need to do a scan of John’s head!



(2)

## Models (the “real worlds”)

Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated

We say  $m$  is a model of a sentence  $\alpha$  if  $\alpha$  is true in  $m$

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

Then  $KB \models \alpha$  if and only if  $M(KB) \subseteq M(\alpha)$

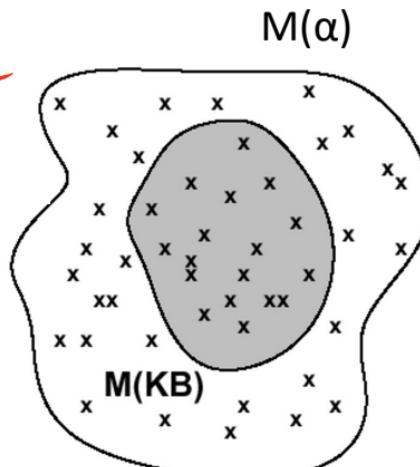
E.g.  $KB = \text{Giants won and Reds won}$

$\alpha = \text{Giants won}$

$\alpha = \text{“intelligence must have a body”}$

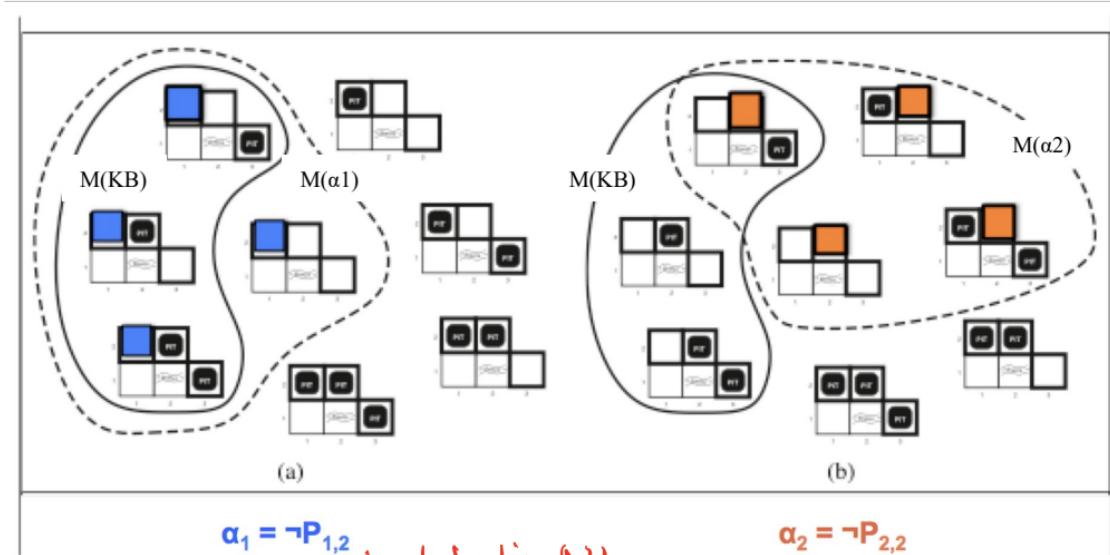
Models (possible worlds):

Earth, Mars, .....



13)

## Check Entailment in Wumpus World



$$\alpha_1 = \neg P_{1,2}$$

no pit at location (1,2)

$$KB \models \alpha_1$$

?  $\top$

$$\alpha_2 = \neg P_{2,2}$$

$$KB \models \alpha_2$$

?  $\perp$

# Inference

from KB, apply procedure i, will get  $\alpha$

$KB \vdash_i \alpha$  = sentence  $\alpha$  can be derived from  $KB$  by procedure  $i$

Soundness:  $i$  is sound if

whenever  $KB \vdash_i \alpha$ , it is also true that  $KB \models \alpha$   $\Rightarrow$  procedure  $i$  is sound,  
but  $\models \vdash \alpha$

$KB$  entails  $\alpha$

Completeness:  $i$  is complete if

whenever  $KB \models \alpha$ , it is also true that  $KB \vdash_i \alpha$   $\Rightarrow$   $KB \models \vdash \alpha$  procedure  $i$  is complete

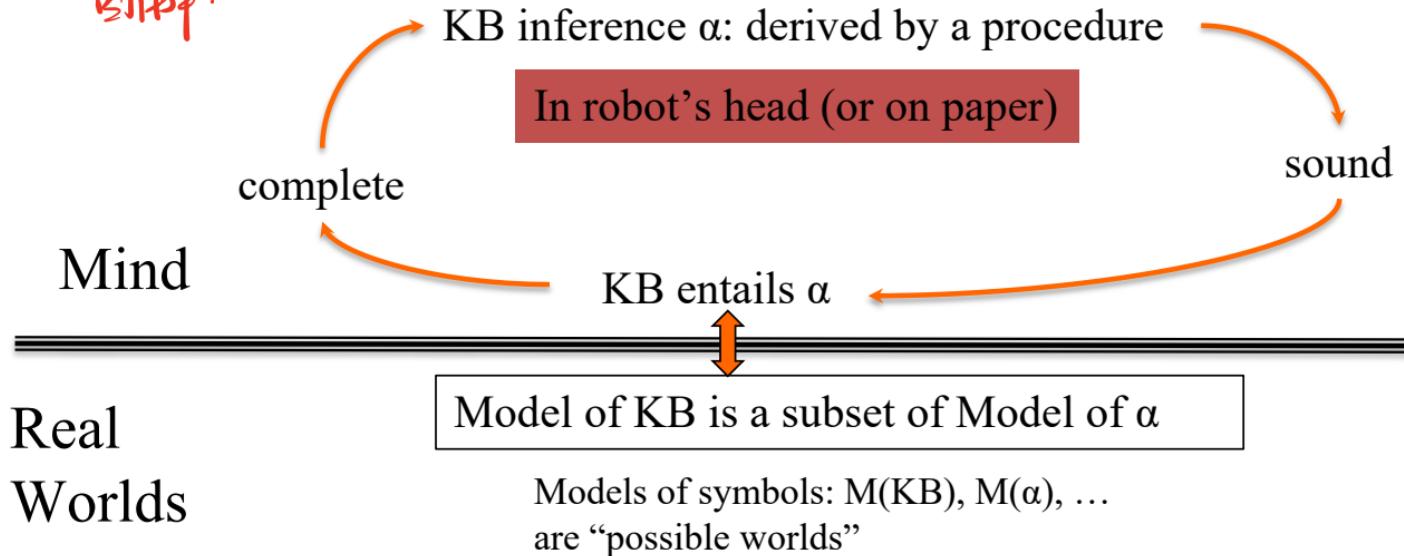
property  
of procedure  $i$

Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the  $KB$ .

# “Entailment” =\= “Inference”

困惑：



三、

## Basic Symbols

1. ~~A~~ Expressions only evaluate to either “true” or “false.”

- $P$  “P is true”
- $\neg P$  “P is false” negation
- $P \vee Q$  “either P is true or Q is true or both” disjunction
- $P \wedge Q$  “both P and Q are true” conjunction
- $P \Rightarrow Q$  “if P is true, then Q is true” implication
- $P \Leftrightarrow Q$  “P and Q are either both true or both false” equivalence

## Propositional Logic: Syntax

Propositional logic is the simplest logic

The proposition symbols  $P_1, P_2$  etc are sentences

If  $S$  is a sentence,  $\neg S$  is a sentence

If  $S_1$  and  $S_2$  is a sentence,  $S_1 \wedge S_2$  is a sentence

If  $S_1$  and  $S_2$  is a sentence,  $S_1 \vee S_2$  is a sentence

If  $S_1$  and  $S_2$  is a sentence,  $S_1 \Rightarrow S_2$  is a sentence

If  $S_1$  and  $S_2$  is a sentence,  $S_1 \Leftrightarrow S_2$  is a sentence

## Propositional Logic: Semantics (must be checked in real worlds)

Each model specifies true/false for each proposition symbol

E.g.     $A$      $B$      $C$   
            *True*    *True*    *False*

Rules for evaluating truth with respect to a model  $m$ :

$\neg S$	is true iff	$S$	is false	
$S_1 \wedge S_2$	is true iff	$S_1$	is true <u>and</u>	$S_2$ is true
$S_1 \vee S_2$	is true iff	$S_1$	is true <u>or</u>	$S_2$ is true
$S_1 \Rightarrow S_2$	is true iff i.e., is false iff	$S_1$	is false <u>or</u>	$S_2$ is true
$S_1 \Leftrightarrow S_2$	is true iff	$S_1$	is true <u>and</u>	$S_2$ is false $\rightarrow$ only

false  $\Rightarrow$  true is valid

$P \Rightarrow Q$  is the same as  $\sim P \vee Q$

## 3. Truth Tables

- Truth value: whether a statement is true or false.
- Truth table: complete list of truth values for a statement given all possible values of the individual atomic expressions.

Example:

P	Q	$\neg P$	$\neg Q$	$P \vee Q$	$P \wedge Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
T	T	F	F	T	T	T	T
T	F	F	T	T	F	F	F
F	T	T	F	T	F	T	F
F	F	T	T	F	F	T	T

## Propositional logic: basic manipulation rules

- $\neg(\neg A) = A$  Double negation
- $\neg(A \wedge B) = (\neg A) \vee (\neg B)$  Negated “and”
- $\neg(A \vee B) = (\neg A) \wedge (\neg B)$  Negated “or”
- $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$  Distributivity of  $\wedge$  on  $\vee$
- $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$  Distributivity of  $\vee$  on  $\wedge$
- $A \Rightarrow B = (\neg A) \vee B$  by definition
- $\neg(A \Rightarrow B) = A \wedge (\neg B)$  using negated or
- $A \Leftrightarrow B = (A \Rightarrow B) \wedge (B \Rightarrow A)$  by definition
- $\neg(A \Leftrightarrow B) = (A \wedge (\neg B)) \vee (B \wedge (\neg A))$  using negated and & or
- ...

## Propositional inference: enumeration method

Let  $\alpha = A \vee B$  and  $KB = (A \vee C) \wedge (B \vee \neg C)$

Is it the case that  $KB \models \alpha$ ?

Check all possible models— $\alpha$  must be true wherever  $KB$  is true

$A$	$B$	$C$	$A \vee C$	$B \vee \neg C$	$KB$	$\alpha$
False	False	False	False	True	False	False
False	False	True	True	False	False	False
False	True	False	False	True	False	True
False	True	True	True	True	True	True
True	False	False	True	True	True	True
True	False	True	True	False	False	True
True	True	False	True	True	True	True
True	True	True	True	True	True	True

$M(KB)$

$M(\alpha)$

4.

## Propositional Logic: Normal Forms

Other approaches to inference use syntactic operations on sentences, often expressed in standardized forms

### (1) Conjunctive Normal Form (CNF—universal)

*conjunction of disjunctions of literals  
  clauses*

E.g.,  $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

"product of sums of simple variables or negated simple variables"

*many class, In class is or, between class is and*

### (2) Disjunctive Normal Form (DNF—universal)

*disjunction of conjunctions of literals  
  terms*

E.g.,  $(A \wedge B) \vee (A \wedge \neg C) \vee (A \wedge \neg D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge \neg D)$

"sum of products of simple variables or negated simple variables"

*many class, In class is and, between class is or*

### (3) Horn Form (restricted)

*conjunction of Horn clauses (clauses with  $\leq 1$  positive literal)*

E.g.,  $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Often written as set of implications:

$B \Rightarrow A$  and  $(C \wedge D) \Rightarrow B$

## 5.

## Deriving expressions from functions

*truth table  $\rightarrow$  logic expression*

- Given a Boolean function in truth table form, find a propositional logic expression for it that uses only  $V$ ,  $\wedge$  and  $\neg$ .
- Idea:** We can easily do it by disjoining the "T" rows of the truth table.

Example: Given a truth table for a function, how to write the results in logic?

P	Q	RESULT
T	T	F
T	F	T
F	T	T
F	F	F

*look at true value row*

$$\begin{aligned} & P \wedge (\neg Q) \\ & (\neg P) \wedge Q \end{aligned}$$

$$\text{RESULT} = (P \wedge (\neg Q)) \vee ((\neg P) \wedge Q)$$

## A more formal approach

- To construct a logical expression in disjunctive normal form from a truth table:
- Build a “**minterm**” for each row of the table, where:
  - For each variable whose value is T in that row, include the variable in the minterm
  - For each variable whose value is F in that row, include the negation of the variable in the minterm
- Link variables in minterm by conjunctions
- The expression consists of the disjunction of all minterms.

## Example: adder with carry

Takes 3 variables in:  $x$ ,  $y$  and  $ci$  (carry-in); yields 2 results: sum ( $s$ ) and carry-out ( $co$ ). To get you used to other notations, here we assume  $T = 1$ ,  $F = 0$ ,  $V = OR$ ,  $\wedge = AND$ ,  $\neg = NOT$ .

$x$	$y$	$ci$	$co$	$s$	
0	0	0	0	0	
0	0	1	0	1	$s : \neg x \wedge \neg y \wedge ci$
0	1	0	0	1	$s : \neg x \wedge y \wedge \neg ci$
0	1	1	1	0	$co : \neg x \wedge y \wedge ci$
1	0	0	0	1	$s : x \wedge \neg y \wedge \neg ci$
1	0	1	1	0	$co : x \wedge \neg y \wedge ci$
1	1	0	1	0	$co : x \wedge y \wedge \neg ci$
1	1	1	1	1	$co, s : x \wedge y \wedge ci$

The logical expression for  $co$  is:

$$(\neg x \wedge y \wedge ci) \vee (x \wedge \neg y \wedge ci) \vee \\ (\neg x \wedge y \wedge \neg ci) \vee (x \wedge y \wedge ci)$$

The logical expression for  $s$  is:

$$(\neg x \wedge \neg y \wedge ci) \vee (\neg x \wedge y \wedge \neg ci) \\ \vee (x \wedge \neg y \wedge \neg ci) \vee (x \wedge y \wedge ci)$$

## b. Tautologies

- Logical expressions that are always true. Can be simplified out.

Examples:

T

$T \vee A$

$A \vee (\neg A)$

$\neg(A \wedge (\neg A))$

$A \Leftrightarrow A$

$((P \vee Q) \Leftrightarrow P) \vee (\neg P \wedge Q)$

$(P \Leftrightarrow Q) \Rightarrow (P \Rightarrow Q)$

## Validity and Satisfiability

(1) A sentence is valid if it is true in all models

e.g.,  $A \vee \neg A$ ,  $A \Rightarrow A$ ,  $(A \wedge (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem

$KB \models \alpha$  if and only if  $(KB \Rightarrow \alpha)$  is valid

(2) A sentence is satisfiable if it is true in some model

e.g.,  $A \vee B$ ,  $C$

(3) A sentence is unsatisfiable if it is true in no models

e.g.,  $A \wedge \neg A$

Satisfiability is connected to inference via the following:

$KB \models \alpha$  if and only if  $(KB \wedge \neg \alpha)$  is unsatisfiable

i.e., prove  $\alpha$  by *reductio ad absurdum*

## 7. Proof methods

Proof methods divide into (roughly) two kinds:

① types:

① Model checking

truth table enumeration (sound and complete for propositional)

heuristic search in model space (sound but incomplete)

e.g., the GSAT algorithm (Ex. 6.15)

② Application of inference rules

Legitimate (sound) generation of new sentences from old

Proof = a sequence of inference rule applications

Can use inference rules as operators in a standard search alg.

8.

## Inference Rules (1-4)

Prove from top to bottom

- ① **Modus Ponens** or **Implication-Elimination**: (From an implication and the premise of the implication, you can infer the conclusion.)

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

- ② **And-Elimination**: (From a conjunction, you can infer any of the conjuncts.)

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n}{\alpha_i}$$

- ③ **And-Introduction**: (From a list of sentences, you can infer their conjunction.)

$$\frac{\alpha_1, \alpha_2, \dots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n}$$

- ④ **Or-Introduction**: (From a sentence, you can infer its disjunction with anything else at all.)

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \dots \vee \alpha_n}$$

## Inference Rules (5-7)

- 5 Double-Negation Elimination: (From a doubly negated sentence, you can infer a positive sentence.)

$$\frac{\neg\neg\alpha}{\alpha}$$

- 6 Unit Resolution: (From a disjunction, if one of the disjuncts is false, then you can infer the other one is true.)

$$\frac{\alpha \vee \beta, \quad \neg\beta}{\alpha}$$

- 7 Resolution: (This is the most difficult. Because  $\beta$  cannot be both true and false, one of the other disjuncts must be true in one of the premises. Or equivalently, implication is transitive.)

$$\frac{\alpha \vee \beta, \quad \neg\beta \vee \gamma}{\alpha \vee \gamma} \quad \text{or equivalently} \quad \frac{\neg\alpha \Rightarrow \beta, \quad \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

eg 1:

## Inference Example

- Prove the rule of "Modus Tollens"  $\frac{\alpha \rightarrow \beta, \neg \beta}{\neg \alpha}$

- $\alpha \rightarrow \beta$   *$\alpha, \beta$  因真假及*
- $\neg \alpha \vee \beta$   *$\downarrow \beta$  too double negation*
- $\neg \neg \beta \vee \neg \alpha$
- $\neg \beta \rightarrow \neg \alpha$
- $\neg \beta$
- $\neg \alpha$

## Inference example

Show that the hypotheses:

- It is not sunny this afternoon and it is colder than yesterday.  $\neg s \wedge c$
- We will go swimming only if it is sunny.  $w \rightarrow s$
- If we do not go swimming, then we will take a canoe trip.  $\neg w \rightarrow t$
- If we take a canoe trip, then we will be home by sunset.  $t \rightarrow h$

lead to the conclusion:

- We will be home by the sunset.  $h$

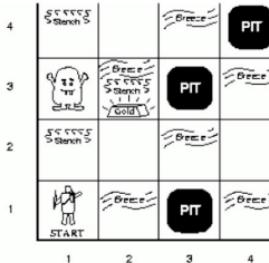
Step	Reason
1. $\neg s \wedge c$	hypothesis
2. $\neg s$	simplification
3. $w \rightarrow s$	hypothesis
4. $\neg w$	modus tollens of 2 and 3
5. $\neg w \rightarrow t$	hypothesis
6. $t$	modus ponens of 4 and 5
7. $t \rightarrow h$	hypothesis
8. $h$	modus ponens of 6 and 7

Where:

- $s$ : "it is sunny this afternoon"  
 $c$ : "it is colder than yesterday"  
 $w$ : "we will go swimming"  
 $t$ : "we will take a canoe trip."  
 $h$ : "we will be home by the sunset."

## Wumpus world: example

- **Facts:** Percepts inject (TELL) facts into the KB
  - [stench at 1,1 and 2,1]  $\square S1,1 ; S2,1$
- **Rules:** if square has no stench then neither the square or adjacent squares contain the Wumpus
  - R1:  $\neg S1,1 \Rightarrow \neg W1,1 \wedge \neg W1,2 \wedge \neg W2,1$
  - R2:  $\neg S2,1 \Rightarrow \neg W1,1 \wedge \neg W2,1 \wedge \neg W2,2 \wedge \neg W3,1$
  - ...
- **Inference:**
  - KB contains  $\neg S1,1$  then using Modus Ponens we infer  
 $\neg W1,1 \wedge \neg W1,2 \wedge \neg W2,1$
  - Using And-Elimination we get:  $\neg W1,1 \quad \neg W1,2 \quad \neg W2,1$
  - ...



## Limitations of Propositional Logic

1. It is too weak, i.e., has very limited expressiveness:

- Each rule has to be represented for each situation:  
e.g., "don't go forward if the Wumpus is in front of you" takes 64 rules

2. It cannot keep track of changes:

- If one needs to track changes, e.g., where the agent has been before then we need a timed-version of each rule. To track 100 steps we'll then need 6400 rules for the previous example.

Its **hard to write and maintain** such a huge rule-base  
**Inference becomes intractable**

## Summary

Logical agents apply inference to a knowledge base to derive new information and make decisions

Basic concepts of logic:

- syntax: formal structure of sentences
- semantics: truth of sentences wrt models
- entailment: necessary truth of one sentence given another
- inference: deriving sentences from other sentences
- soundness: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Propositional logic suffices for some of these tasks

Truth table method is sound and complete for propositional logic

## Summary: Logic and Reasoning

- Knowledge Base (KB): contains a set of sentences expressed using a Knowledge Representation Language (KRL)
  - TELL: operator to add a sentence to the KB
  - ASK: to query the KB
- Logics are KRLs where conclusions can be drawn
  - Syntax
  - Semantics
- Entailment:  $\text{KB} \models a$  iff “ $a$  is true in all worlds where KB is true”
- Inference:  $\text{KB} \vdash_i a$  = sentence  $a$  can be derived from KB using a procedure  $i$ 
  - Sound: whenever  $\text{KB} \vdash_i a$ , then  $\text{KB} \models a$  is true
  - Complete: whenever  $\text{KB} \models a$ , then  $\text{KB} \vdash_i a$