# Knowledge-Based Agents

# Big Idea

- 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 programs 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

# Inference in People

- People can do logical inference, but are not very good at it.
- Reasoning with negation and disjunction seems to be particularly difficult.
- But, people seem to employ many kinds of reasoning strategies, most of which are neither complete nor sound.

# Wason Selection Task

- I have a pack of cards each of which has a letter written on one side and a number written on the other side and I claim the following rule is true:

  *If a card has a vowel on one side, then it has an even number on the other side.*

- Look at these cards and say which card or cards to turn over in order to decide whether the rule is true or false?

| | | | |
|---|---|---|---|
| **E** | **4** | **T** | **7** |

# Wason Selection Task

- Wason (1966) showed that people are not very good at this task.

  Wason, P. (1966). Reasoning. In New Horizons in Psychology. Penguin, Hammondsworth, UK.

- To disprove P=>Q, find a situation in which P^~Q
- To disprove vowel => even, find a card with a vowel and an odd number
- Thus, turn over the cards showing vowels and turn over cards showing odd numbers

# Wason Selection Task

- This version seems easier for people to do, as was shown by Griggs & Cox, 1982
- Your are the bouncer in a bar.  Which of these people do you card given the rule:

  *You must be 21 or older to drink beer.*

| | | | |
|---|---|---|---|
| **beer** | **coke** | **22** | **20** |

- It may be simpler because it's more familiar or because people have special strategies to reason about certain situations, such as "cheating" in a social situation.

# Logic as a Methodology

Even if people do not use formal logical reasoning for solving a problem, logic might be a good approach for AI for a number of reasons

- Airplanes don’t need to flap their wings
- Logic may be a good implementation strategy
- Developing a solution in a formal system like logic can offer other benefits, e.g., letting us prove properties of the approach

# A knowledge-based agent

- A knowledge-based agent includes a knowledge base and an inference system.
- A knowledge base is a set of representations of facts of the world.
- Each individual representation is called a **sentence**.
- The sentences are expressed in a **knowledge representation language**.
- The agent operates as follows:
    1. It TELLs the knowledge base what it perceives.
    2. It ASKs the knowledge base what action it should perform.
    3. It performs the chosen action.

# Architecture of a KB agent

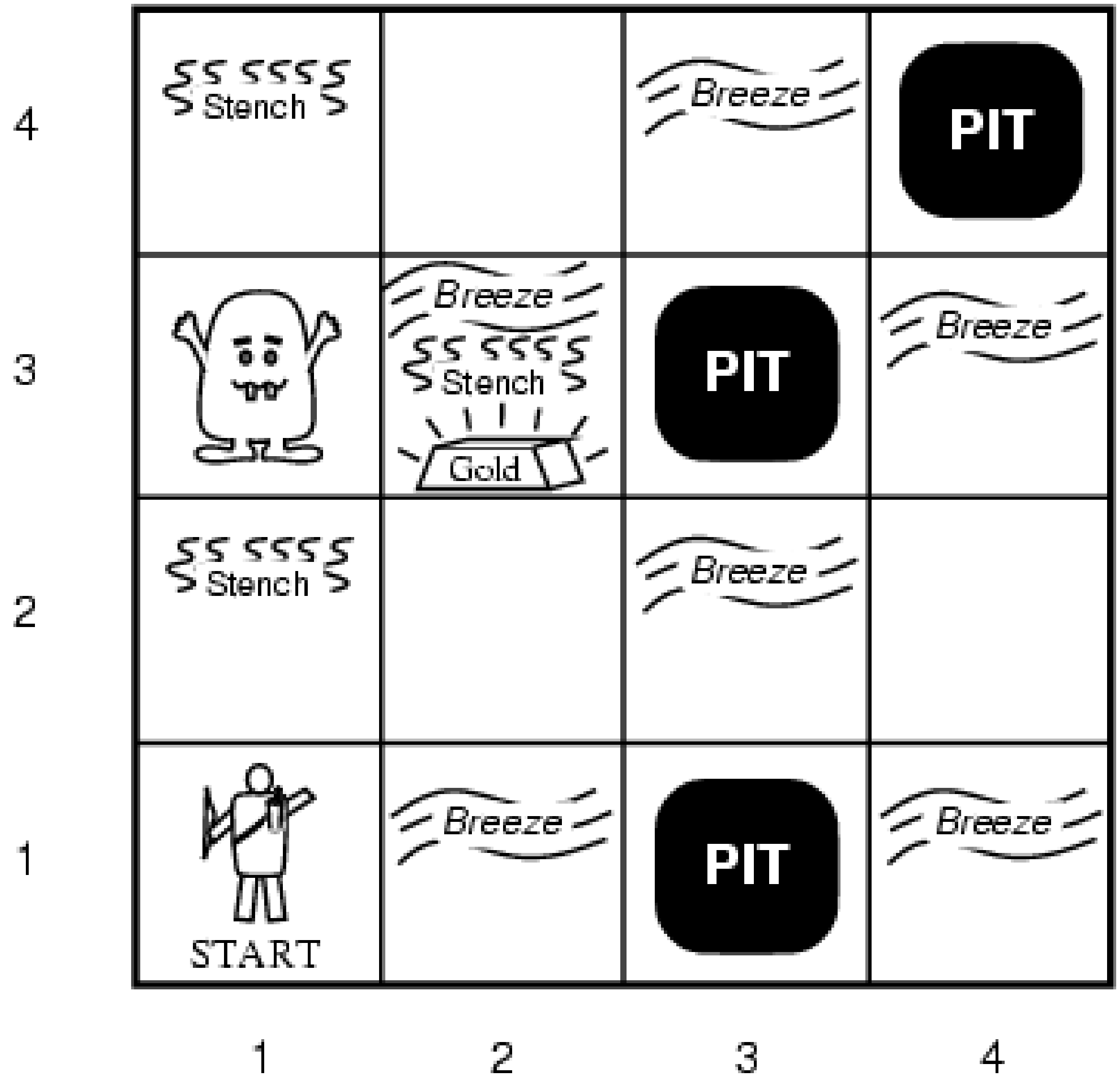
- **Knowledge Level**
  - The most abstract level: describe agent by saying what it knows
  - Ex: A taxi agent might know that the Golden Gate Bridge connects San Francisco with the Marin County
- **Logical Level**
  - The level at which the knowledge is encoded into *sentences*
  - Ex: links(GoldenGateBridge, SanFrancisco, MarinCounty)
- **Implementation Level**
  - The physical representation of the sentences in the logical level
  - Ex: `‘(links goldengatebridge sanfrancisco marincounty)`

# The Wumpus World environment

- The Wumpus computer game
- The agent explores a cave consisting of rooms connected by passageways
- Lurking somewhere in the cave is the Wumpus, a beast that eats any agent that enters its room
- Some rooms contain bottomless pits that trap any agent that wanders into the room
- The Wumpus can fall into a pit too, so avoids them
- Occasionally, there is a heap of gold in a room.
- The goal is to collect the gold and exit the world without being eaten

# AIMA's Wumpus World

- The agent always starts in the field [1,1]
- Agent's task is to find the gold, return to the field [1,1] and climb out of the cave



# Agent in a Wumpus world: Percepts

- The agent perceives
  - a **stench** in the square containing the Wumpus and in the adjacent squares (not diagonally)
  - a **breeze** in the squares adjacent to a pit
  - a **glitter** in the square where the gold is
  - a **bump**, if it walks into a wall
  - a woeful **scream** everywhere in the cave, if the Wumpus is killed
- The percepts are given as a five-symbol list. If there is a stench and a breeze, but no glitter, no bump, and no scream, the percept is

  [Stench, Breeze, None, None, None]
- The agent cannot perceive its own location

# Wumpus World Actions

- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab**: Pick up an object that’s in the same square as the agent
- **shoot**: Fire an arrow in a straight line in the direction the agent is facing. It continues until it hits and kills the Wumpus or hits the outer wall. The agent has only one arrow, so only the first shoot action has any effect
- **climb** is used to leave the cave and is only effective in the start square
- **die**: This action automatically and irretrievably happens if the agent enters a square with a pit or a live Wumpus

# Wumpus World Goal

The agent's goal is to find the gold and bring it back to the start square as quickly as possible, without getting killed

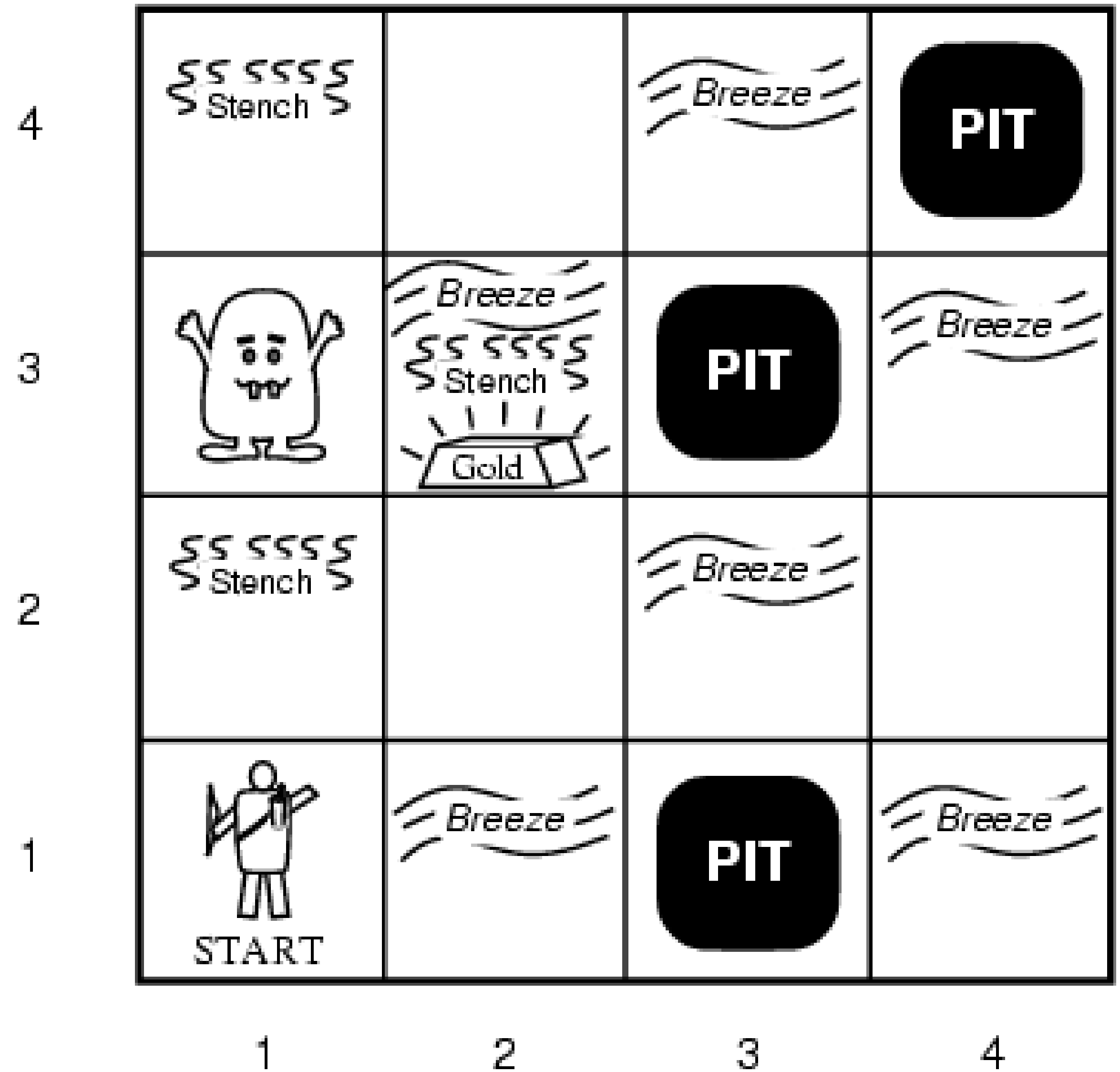
- 1,000 points reward for climbing out of the cave with the gold
- 1 point deducted for every action taken
- 10,000 points penalty for getting killed

# Wumpus world characterization

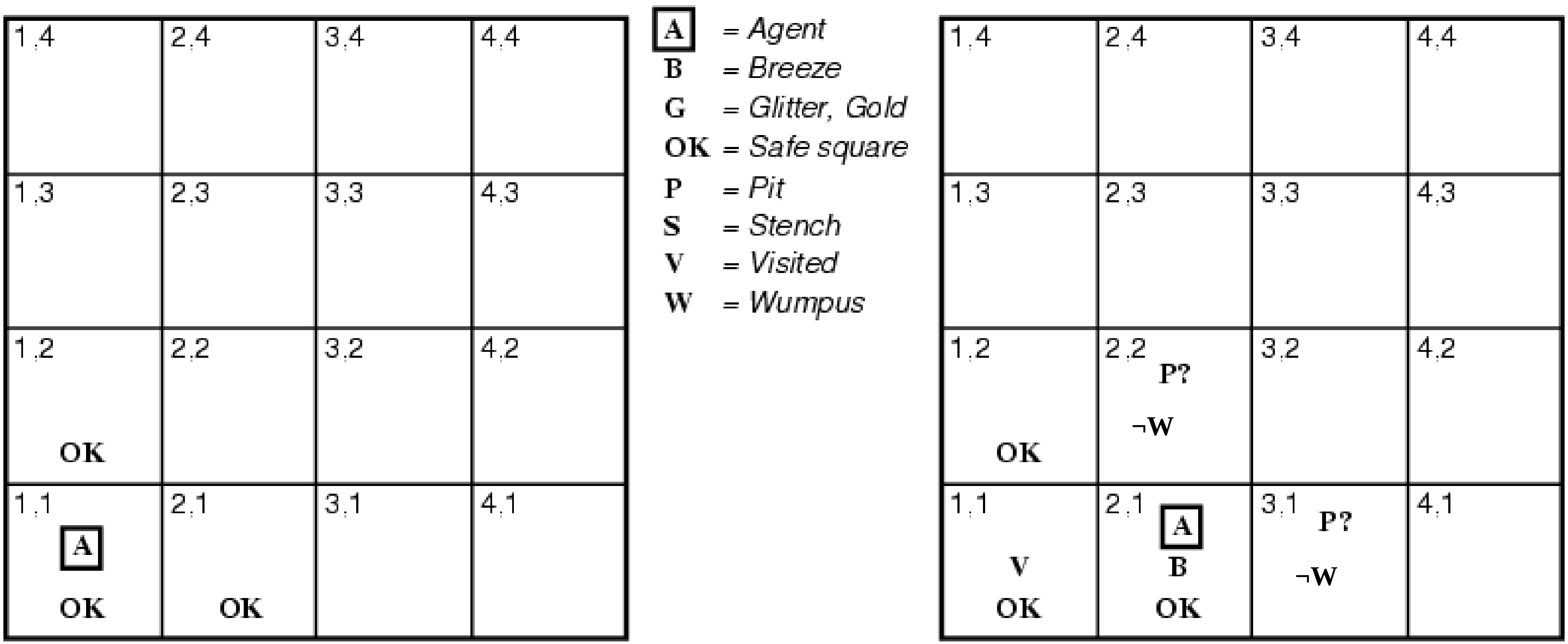
- **Fully Observable** No – only local perception
- **Deterministic** Yes – outcomes exactly specified
- **Episodic** No – sequential at the level of actions
- **Static** Yes – Wumpus and Pits do not move
- **Discrete** Yes
- **Single-agent?** Yes – Wumpus is essentially a natural feature

# AIMA's Wumpus World

- The agent always starts in the field [1,1]
- Agent's task is to find the gold, return to the field [1,1] and climb out of the cave

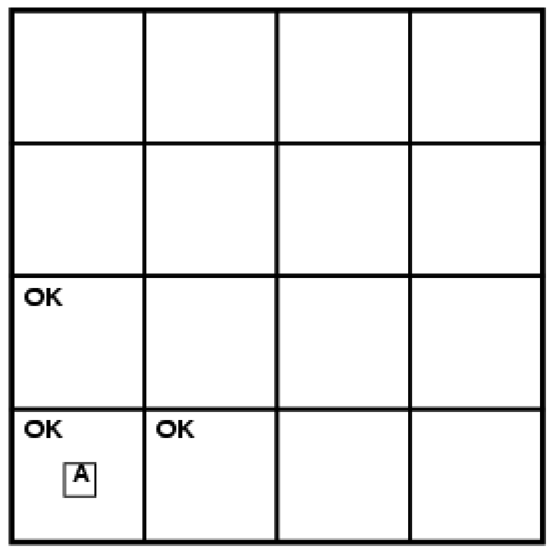


# The Hunter’s first step



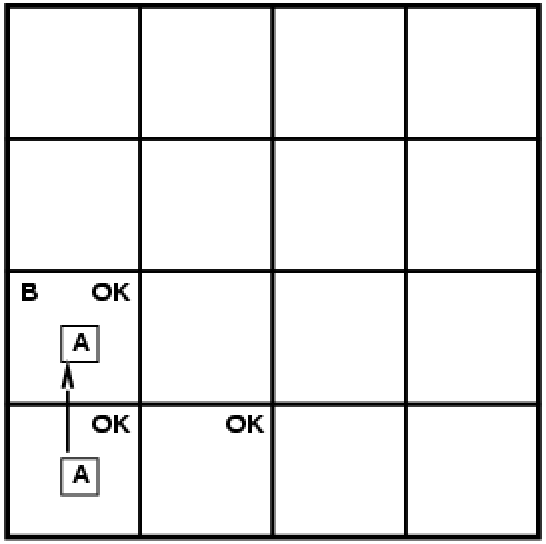

(a) (b)

# Exploring a wumpus world



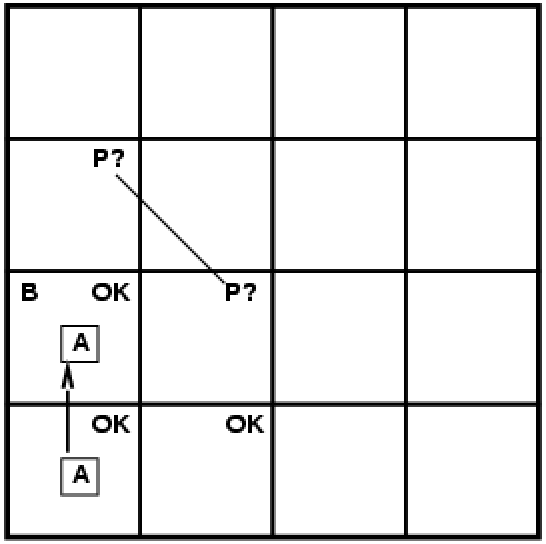

| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world



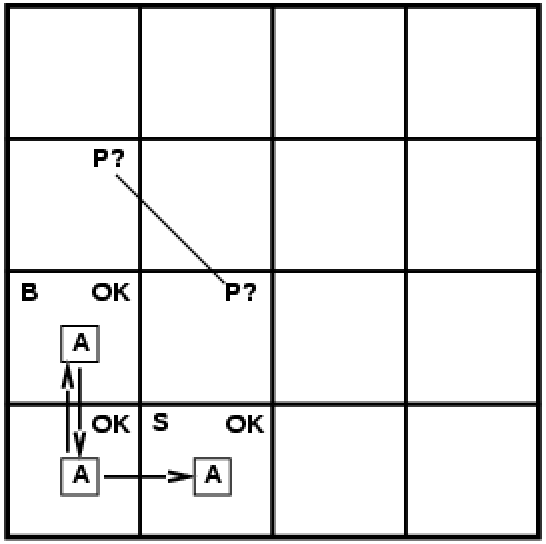

| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world



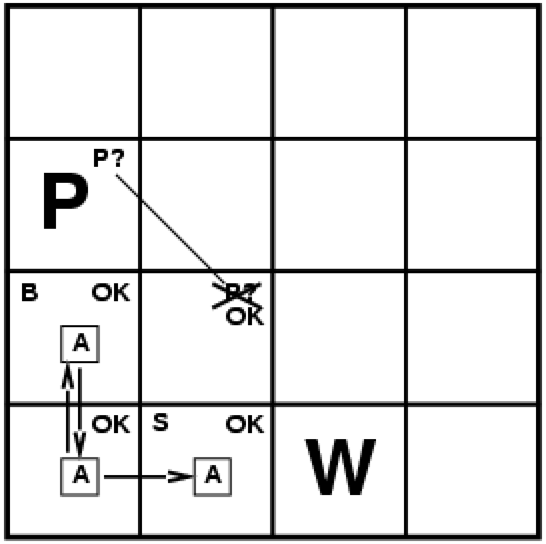

| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world



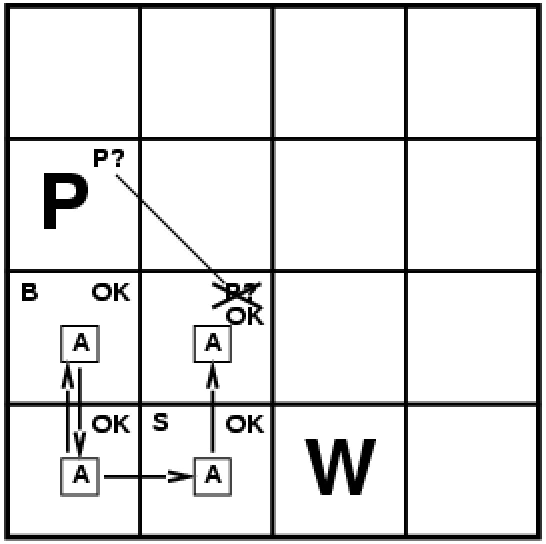

| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world



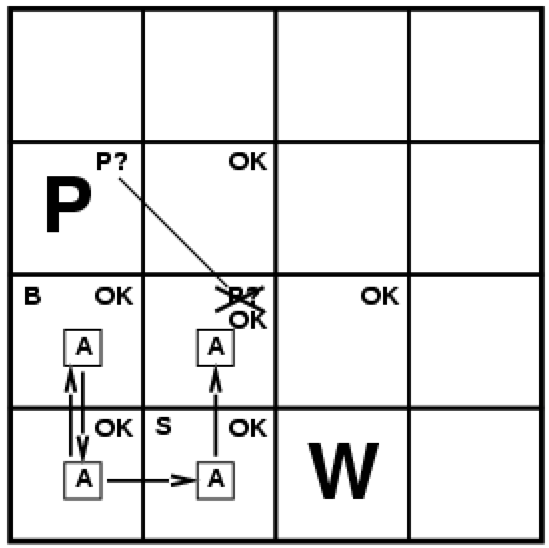

| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world



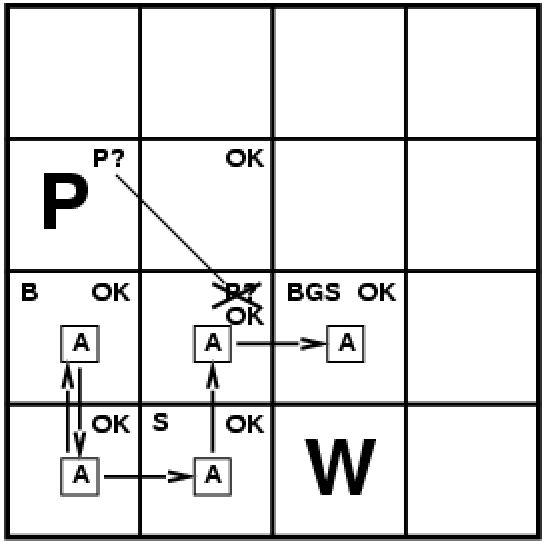

| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world



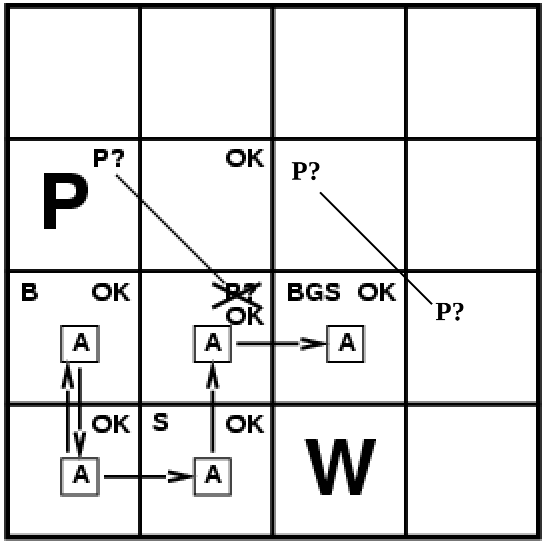

| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world




| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Exploring a wumpus world




| | |
|---|---|
| A | agent |
| B | breeze |
| G | glitter |
| OK | safe cell |
| P | pit |
| S | stench |
| W | wumpus |

# Logic in general

- **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 $x = 0$, $y = 6$

# Entailment

- Entailment 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

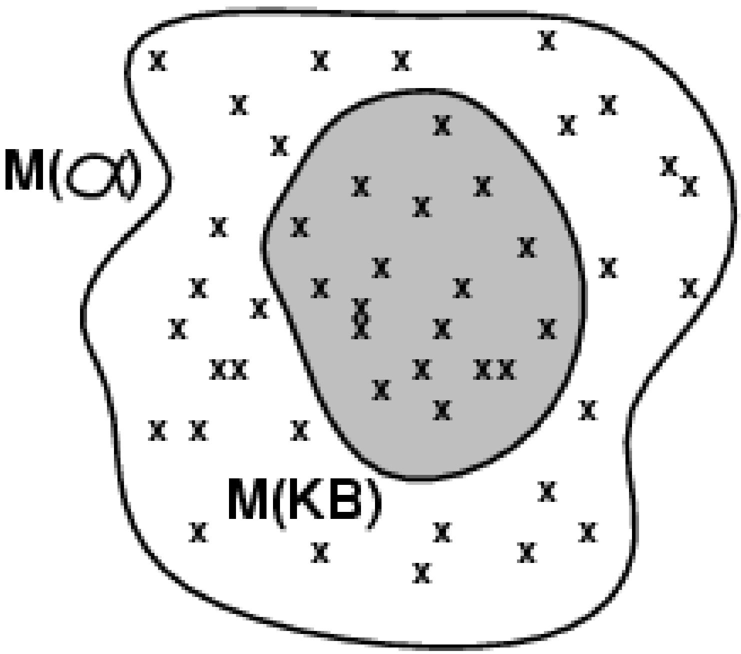
  – Entailment is a relationship between sentences (i.e., syntax) that is based on **semantics**

# Models

- 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 α if α is true in *m*

- *M(α)* is the set of all models of α



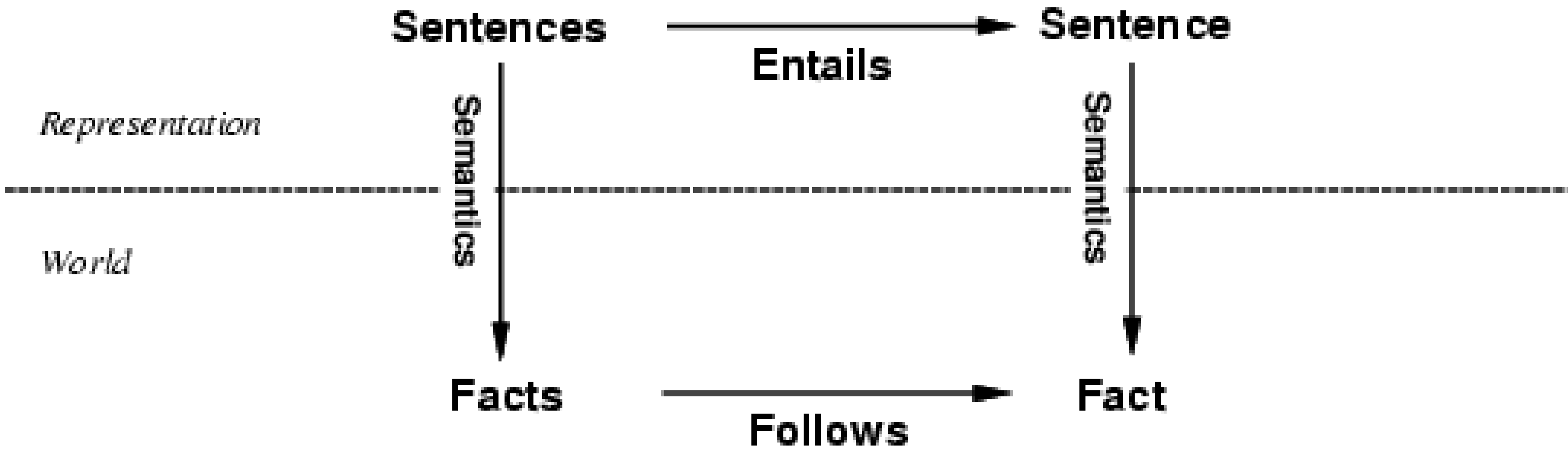
# Inference, Soundness, Completeness

- $KB \vdash_i \alpha$ = 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 $KB \models \alpha$
- **Completeness:** *i* is complete if whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$
- 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*.

# Representation, reasoning, and logic

- The object of knowledge representation is to express knowledge in a **computer-tractable** form, so that agents can perform well
- A knowledge representation language is defined by:
  - its **syntax,** which defines all possible sequences of symbols that constitute sentences of the language.
    - Ex: Sentences in a book, bit patterns in computer memory
  - its **semantics**, which determines the facts in the world to which the sentences refer.
    - Each sentence makes a claim about the world.
    - An agent is said to believe a sentence about the world.

# The connection between sentences and facts




*Semantics* maps sentences in logic to facts in the world. The property of one fact following from another is mirrored by the property of one sentence being entailed by another.

# Soundness and completeness

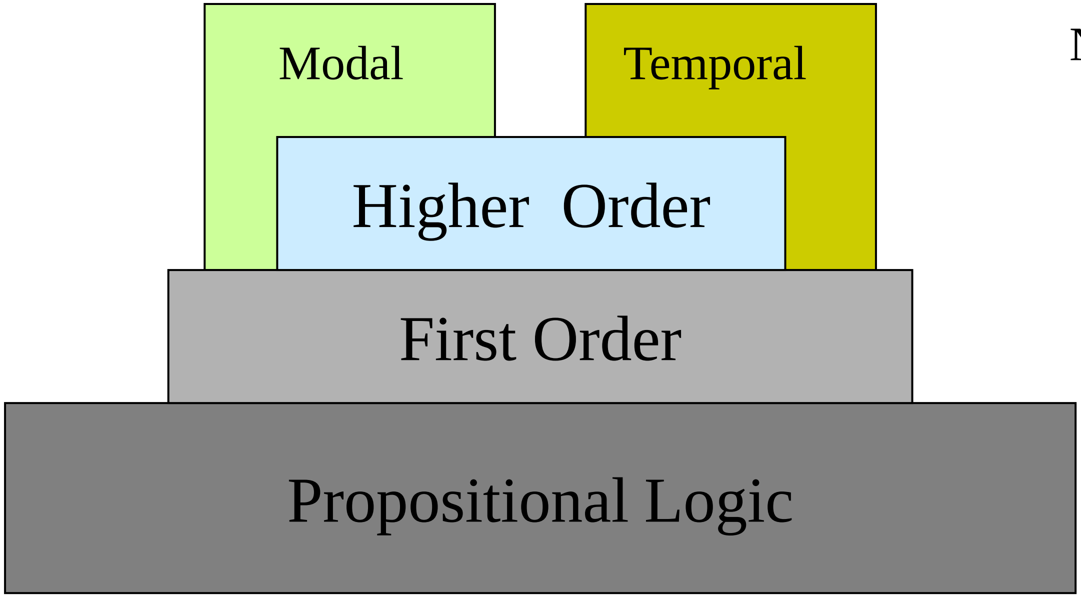
- A *sound* inference method derives only entailed sentences
- Analogous to the property of *completeness* in search, a *complete* inference method can derive any sentence that is entailed

# Logic as a KR language

Multi-valued Logic

Probabilistic Logic

Fuzzy Logic




Non-monotonic Logic

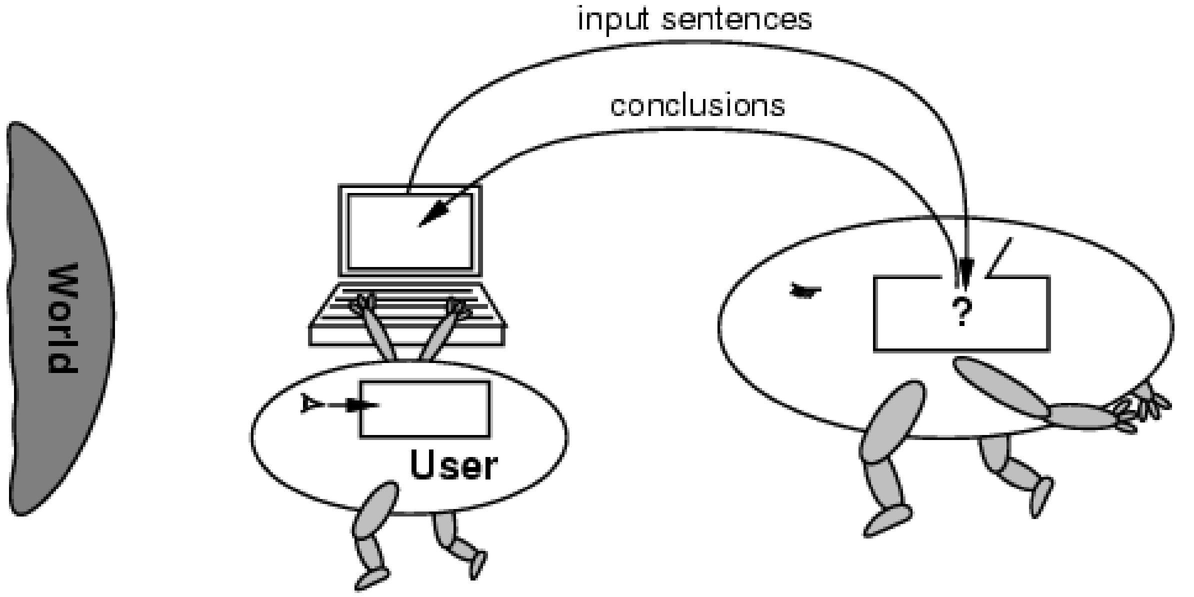
# Ontology and epistemology

- **Ontology** is the study of what there is—an inventory of what exists. An ontological commitment is a commitment to an existence claim.
- **Epistemology** is a major branch of philosophy that concerns the forms, nature, and preconditions of knowledge.

| Language | Ontological Commitment (What exists in the world) | Epistemological Commitment (What an agent believes about facts) |
|---|---|---|
| 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 $0 \ldots 1$ |
| Fuzzy logic | degree of truth | degree of belief $0 \ldots 1$ |

# No independent access to the world

- The reasoning agent often gets its knowledge about the facts of the world as a sequence of logical sentences and must draw conclusions only from them without independent access to the world.
- Thus, it is very important that the agent's reasoning is sound!



# Logic roadmap overview

- Propositional logic (review)
- Problems with propositional logic
- First-order logic (review)
  - Properties, relations, functions, quantifiers, …
  - Terms, sentences, wffs, axioms, theories, proofs, …
- Extensions to first-order logic
- Logical agents
  - Reflex agents
  - Representing change: situation calculus, frame problem
  - Preferences on actions
  - Goal-based agents

# Disclaimer

“Logic, like whiskey, loses its beneficial effect when taken in too large quantities.”

- *Lord Dunsany*

# Big Ideas

- Logic is a great knowledge representation language for many AI problems
- **Propositional logic** is the simple foundation and fine for some AI problems
- **First order logic** (FOL) is much more expressive as a KR language and more commonly used in AI
- There are many variations: horn logic, higher order logic, three-valued logic, probabilistic logics, etc.

# Propositional logic

- **Logical constants**: true, false
- **Propositional symbols**: P, Q,... (**atomic sentences**)
- Wrapping **parentheses**: ( … )
- Sentences are combined by **connectives**:

| | | |
|---|---|---|
| $\wedge$ | and | [conjunction] |
| $\vee$ | or | [disjunction] |
| $\Rightarrow$ | implies | [implication / conditional] |
| $\Leftrightarrow$ | is equivalent | [biconditional] |
| $\neg$ | not | [negation] |

- **Literal**: atomic sentence or negated atomic sentence

  P, $\neg$ P

# Examples of PL sentences

- $(P \wedge Q) \rightarrow R$

  "If it is hot and humid, then it is raining"
- $Q \rightarrow P$

  "If it is humid, then it is hot"
- $Q$

  "It is humid."
- We're free to choose better symbols, btw:

  Ho = "It is hot"

  Hu = "It is humid"

  R = "It is raining"

# Propositional logic (PL)

- Simple language for showing key ideas and definitions
- User defines set of propositional symbols, like P and Q
- User defines **semantics** of each propositional symbol:
  - P means “It is hot”, Q means “It is humid”, etc.
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg S$ is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then $(S \vee T)$, $(S \wedge T)$, $(S \rightarrow T)$, and $(S \leftrightarrow T)$ are sentences
  - A sentence results from a finite number of applications of the rules

# Some terms

- The meaning or **semantics** of a sentence determines its **interpretation**
- Given the truth values of all symbols in a sentence, it can be “evaluated” to determine its **truth value** (True or False)
- A **model** for a KB is a *possible world* – an assignment of truth values to propositional symbols that makes each sentence in the KB True

# More terms

- A **valid sentence** or **tautology** is a sentence that is True under all interpretations, no matter what the world is actually like or what the semantics is. Example: “It’s raining or it’s not raining”
- An **inconsistent sentence** or **contradiction** is a sentence that is False under all interpretations. The world is never like what it describes, as in “It’s raining and it’s not raining.”
- **P entails Q**, written P |= Q, means that whenever P is True, so is Q. In other words, all models of P are also models of Q.

# Truth tables

- Truth tables are used to define logical connectives
- and to determine when a complex sentence is true given the values of the symbols in it

*Truth tables for the five logical connectives*

| $P$ | $Q$ | $\neg P$ | $P \wedge Q$ | $P \vee Q$ | $P \Rightarrow Q$ | $P \Leftrightarrow Q$ |
|---|---|---|---|---|---|---|
| *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* |

*Example of a truth table used for a complex sentence*

| $P$ | $H$ | $P \vee H$ | $(P \vee H) \wedge \neg H$ | $((P \vee H) \wedge \neg H) \Rightarrow P$ |
|---|---|---|---|---|
| *False* | *False* | *False* | *False* | *True* |
| *False* | *True* | *True* | *False* | *True* |
| *True* | *False* | *True* | *True* | *True* |
| *True* | *True* | *True* | *False* | *True* |

# On the implies connective: $P \rightarrow Q$

- Note that $\rightarrow$ is a logical connective
- So $P \rightarrow Q$ is a logical sentence and has a truth value, i.e., is either true or false
- If we add this sentence to the KB, it can be used by an inference rule, *Modes Ponens*, to derive/infer/prove Q if P is also in the KB
- Given a KB where P=True and Q=True, we can also derive/infer/prove that $P \rightarrow Q$ is True

# $\mathbf{P \to Q}$

- When is $P \to Q$ true? Check all that apply
  - ❑ P=Q=true
  - ❑ P=Q=false
  - ❑ P=true, Q=false
  - ❑ P=false, Q=true

# $\mathbf{P \rightarrow Q}$

- When is $P \rightarrow Q$ true? Check all that apply
  - ☑ P=Q=true
  - ☑ P=Q=false
  - ☐ P=true, Q=false
  - ☑ P=false, Q=true
- We can get this from the truth table for $\rightarrow$
- Note: in FOL it's much harder to prove that a conditional true.
  - Consider proving prime(x) $\rightarrow$ odd(x)

# Inference rules

- **Logical inference** creates new sentences that logically follow from a set of sentences (KB)
- An inference rule is **sound** if every sentence X it produces when operating on a KB logically follows from the KB
  - i.e., inference rule creates no contradictions
- An inference rule is **complete** if it can produce every expression that logically follows from (is entailed by) the KB.
  - Note analogy to complete search algorithms

# Exercises

Interpret the following models.

a. $(\alpha \vee \beta) \wedge (\beta \vee \gamma) \leftrightarrow (\alpha \vee \gamma)$

b. $(P \vee H) \wedge -H) \rightarrow P$

# Sound rules of inference

- Here are some examples of sound rules of inference
- Each can be shown to be sound using a truth table

| RULE | PREMISE | CONCLUSION |
| --- | --- | --- |
| Modus Ponens | $A, A \rightarrow B$ | $B$ |
| And Introduction | $A, B$ | $A \wedge B$ |
| And Elimination | $A \wedge B$ | $A$ |
| Double Negation | $\neg\neg A$ | $A$ |
| Unit Resolution | $A \vee B, \neg B$ | $A$ |
| **Resolution** | $\mathbf{A \vee B, \neg B \vee C}$ | $\mathbf{A \vee C}$ |

# Resolution

- **Resolution** is a valid inference rule producing a new clause implied by two clauses containing *complementary literals*
  - A literal is an atomic symbol or its negation, i.e., P, ~P
- Amazingly, this is the only interference rule you need to build a sound and complete theorem prover
  - Based on proof by contradiction and usually called resolution refutation
- The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 60s

# Resolution

- A KB is actually a set of sentences all of which are true, i.e., a conjunction of sentences.
- To use resolution, put KB into *conjunctive normal form* (CNF), where each sentence written as a disjunction of (one or more) literals

**Tautologies**

$(A \rightarrow B) \leftrightarrow (\sim A \vee B)$

$(A \vee (B \wedge C)) \leftrightarrow (A \vee B) \wedge (A \vee C)$

Example

- KB: $[P \rightarrow Q \text{ , } Q \rightarrow R \wedge S]$
- KB in CNF: $[\sim P \vee Q \text{ , } \sim Q \vee R \text{ , } \sim Q \vee S]$
- Resolve KB(1) and KB(2) producing: $\sim P \vee R$ *(i.e., $P \rightarrow R$)*
- Resolve KB(1) and KB(3) producing: $\sim P \vee S$ *(i.e., $P \rightarrow S$)*
- New KB: $[\sim P \vee Q \text{ , } \sim Q \vee \sim R \vee \sim S \text{ , } \sim P \vee R \text{ , } \sim P \vee S]$

# How do we convert a formula into CNF?

1. Elimination of $\rightarrow$ and $\leftrightarrow$ by means of:

• $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$,

• $A \rightarrow B \equiv \neg A \vee B$

2. push $\neg$ inwards by means of

(a) $\neg(A \wedge B) \equiv \neg A \vee \neg B$ (De Morgan)

(b) $\neg(A \vee B) \equiv \neg A \wedge \neg B$ (De Morgan)

(c) $\neg\neg A \equiv A$ (double negation)

3. use the distributive law

$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ to effect the conversation to CNF.

# Proving things

- A **proof** is a sequence of sentences, where each is a premise or is derived from earlier sentences in the proof by an inference rule
- The last sentence is the **theorem** (also called goal or query) that we want to prove
- Example for the "weather problem"

| | | |
|---|---|---|
| 1 Hu | premise | "It's humid" |
| 2 $Hu \rightarrow Ho$ | premise | "If it's humid, it's hot" |
| 3 Ho | modus ponens(1,2) | "It's hot" |
| 4 $(Ho \wedge Hu) \rightarrow R$ | premise | "If it's hot & humid, it's raining" |
| 5 $Ho \wedge Hu$ | and introduction(1,3) | "It's hot and humid" |
| 6 R | modus ponens(4,5) | "It's raining" |

# Horn sentences

- A **Horn sentence** or **Horn clause** has the form:

  $P1 \wedge P2 \wedge P3 \ldots \wedge Pn \rightarrow Qm$ *where n>=0, m in{0,1}*
- Note: a conjunction of 0 or more symbols to left of $\rightarrow$ and 0-1 symbols to right
- Special cases:
  - n=0, m=1: P (assert P is true)
  - n>0, m=0: $P \wedge Q \rightarrow$ (constraint: both P and Q can't be true)
  - n=0, m=0: (well, there is nothing there!)
- Put in CNF: each sentence is a disjunction of literals with at most one non-negative literal

  $\neg P1 \vee \neg P2 \vee \neg P3 \ldots \vee \neg Pn \vee Q$

  ***$(P \rightarrow Q) = (\neg P \vee Q)$***

# Significance of Horn logic

- We can also have horn sentences in FOL
- Reasoning with horn clauses is much simpler
  - Satisfiability of a propositional KB (i.e., finding values for a symbols that will make it true) is NP complete
  - Restricting KB to horn sentences, satisfiability is in P
- For this reason, FOL Horn sentences are the basis for Prolog and Datalog
- What Horn sentences give up are handling, in a general way, (1) negation and (2) disjunctions

# Entailment and derivation

- **Entailment: KB |= Q**
  - Q is entailed by KB (set sentences) iff there is no logically possible world where Q is false while all the sentences in KB are true
  - Or, stated positively, Q is entailed by KB iff the conclusion is true in every logically possible world in which all the premises in KB are true
- **Derivation: KB |- Q**
  - We can derive Q from KB if there's a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

# Two important properties for inference

**Soundness: If KB |- Q then KB |= Q**

- If Q is derived from KB using a given set of rules of inference, then Q is entailed by KB
- Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid

**Completeness: If KB |= Q then KB |- Q**

- If Q is entailed by KB, then Q can be derived from KB using the rules of inference
- Hence, inference produces all entailments, or all valid sentences can be proved from the premises

# Problems with Propositional Logic

# Propositional logic: pro and con

- Advantages
  - Simple KR language sufficient for some problems
  - Lays the foundation for higher logics (e.g., FOL)
  - Reasoning is decidable, though NP complete, and efficient techniques exist for many problems
- Disadvantages
  - Not expressive enough for most problems
  - Even when it is, it can very “un-concise”

# PL is a weak KR language

- Hard to identify "individuals" (e.g., Mary, 3)
- Can't directly talk about properties of individuals or relations between individuals (e.g., "Bill is tall")
- Generalizations, patterns, regularities can't easily be represented (e.g., "all triangles have 3 sides")
- First-Order Logic (FOL) is expressive enough to represent this kind of information using relations, variables and quantifiers, e.g.,
  - *Every elephant is gray:* $\forall$ x (elephant(x) $\rightarrow$ gray(x))
  - *There is a white alligator:* $\exists$ x (alligator(X) ^ white(X))

# PL Example

- Consider the problem of representing the following information:
  - Every person is mortal.
  - Confucius is a person.
  - Confucius is mortal.
- How can these sentences be represented so that we can infer the third sentence from the first two?

# PL Example

- In PL we have to create propositional symbols to stand for all or part of each sentence, e.g.:

  P = “person”; Q = “mortal”; R = “Confucius”
- The above 3 sentences are represented as:

  $P \rightarrow Q$; $R \rightarrow P$; $R \rightarrow Q$
- The 3rd sentence is entailed by the first two, but we need an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes *person* and *mortal*
- Representing other individuals requires introducing separate symbols for each, with some way to represent the fact that all individuals who are “people” are also “mortal”

# Hunt the Wumpus domain

- Some atomic propositions:

  S12 = There is a stench in cell (1,2)

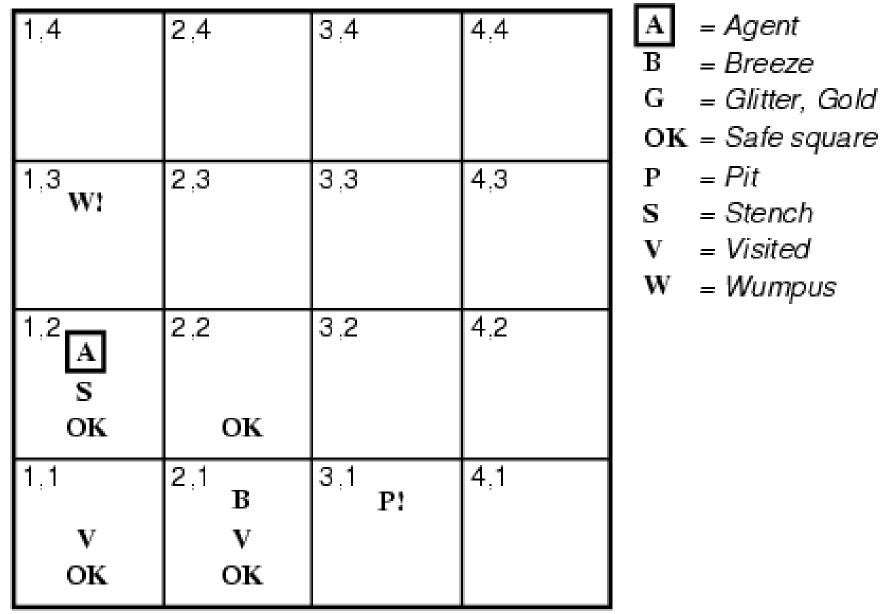
  B34 = There is a breeze in cell (3,4)

  W22 = Wumpus is in cell (2,2)

  V11 = We’ve visited cell (1,1)

  OK11 = Cell (1,1) is safe.

  …




- Some rules:

  *(R1)* $\neg S11 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W21$

  *(R2)* $\neg S21 \rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$

  *(R3)* $\neg S12 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$

  *(R4)* $S12 \rightarrow W13 \vee W12 \vee W22 \vee W11$

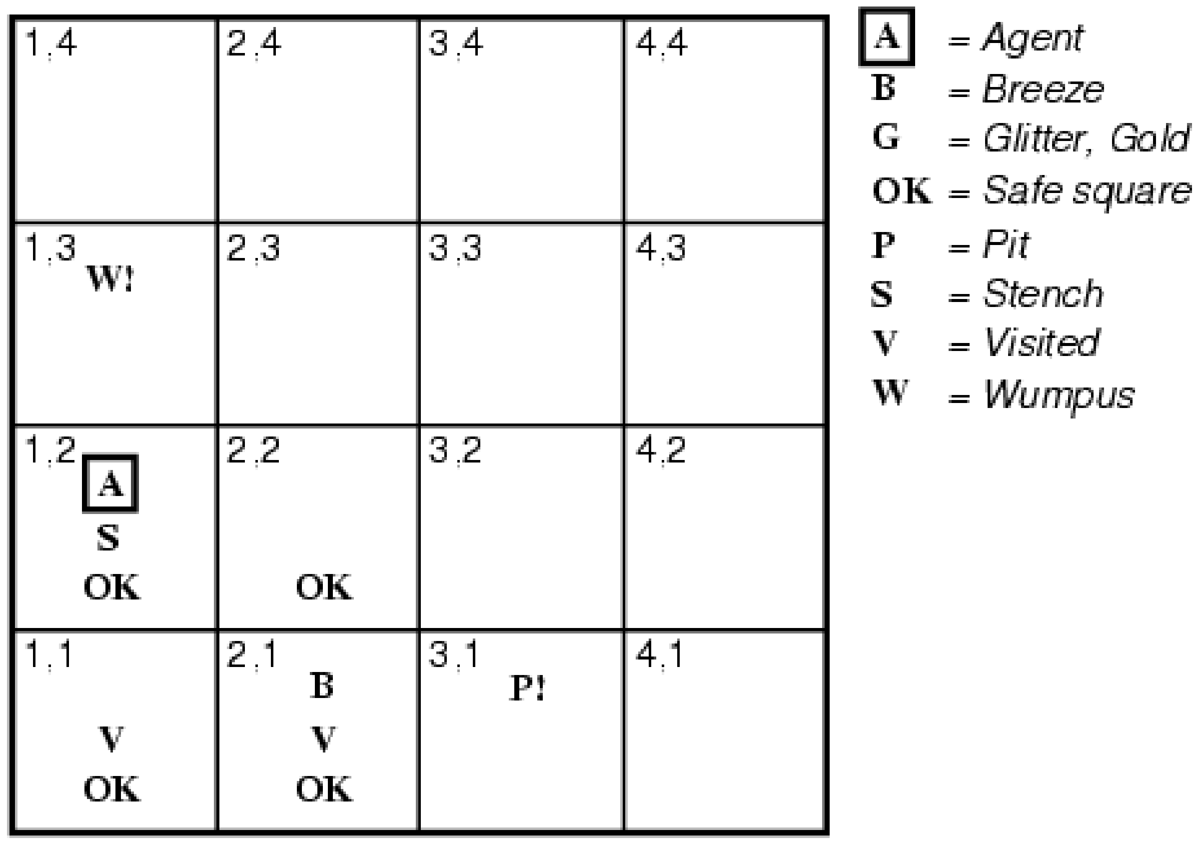
  …

- The lack of variables requires us to give similar rules for each cell!

# After the third move

We can prove that the Wumpus is in (1,3) using the four rules given.

See R&N section 7.5



# Proving W13

Apply MP with $\neg S11$ and R1:

$\neg W11 \wedge \neg W12 \wedge \neg W21$

Apply And-Elimination to this, yielding 3 sentences:

$\neg W11, \neg W12, \neg W21$

Apply MP to ~S21 and R2, then apply And-elimination:

$\neg W22, \neg W21, \neg W31$

Apply MP to S12 and R4 to obtain:

$W13 \vee W12 \vee W22 \vee W11$

Apply Unit resolution on $(W13 \vee W12 \vee W22 \vee W11)$ and $\neg W11$:

$W13 \vee W12 \vee W22$

Apply Unit Resolution with $(W13 \vee W12 \vee W22)$ and $\neg W22$:

$W13 \vee W12$

Apply UR with $(W13 \vee W12)$ and $\neg W12$:

W13

QED

# Propositional Wumpus hunter problems

- Lack of variables prevents stating more general rules
  - We need a set of similar rules for each cell
- Change of the KB over time is difficult to represent
  - Standard technique is to index facts with the time when they're true
  - This means we have a separate KB for every time point

# Propositional logic summary

- Inference is the process of deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives all true conclusions from a set of premises
- A **valid sentence** is true in all worlds under all interpretations
- If an implication sentence can be shown to be valid, then—given its premise—its consequent can be derived
- Different logics make different **commitments** about what the world is made of and what kind of beliefs we can have
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - Simple syntax and semantics suffices to illustrate the process of inference
  - Propositional logic can become impractical, even for very small worlds

# Summary

- Intelligent agents need knowledge about the world for making good decisions.
- The knowledge of an agent is stored in a knowledge base (KB) in the form of **sentences** in a knowledge representation (KR) language.
- A knowledge-based agent needs a **KB** and an **inference mechanism**. It operates by storing sentences in its knowledge base, inferring new sentences with the inference mechanism, and using them to deduce which actions to take.
- A **representation language** is defined by its syntax and semantics, which specify the structure of sentences and how they relate to the facts of the world.
- The **interpretation** of a sentence is the fact to which it refers. If the fact is part of the actual world, then the sentence is true.