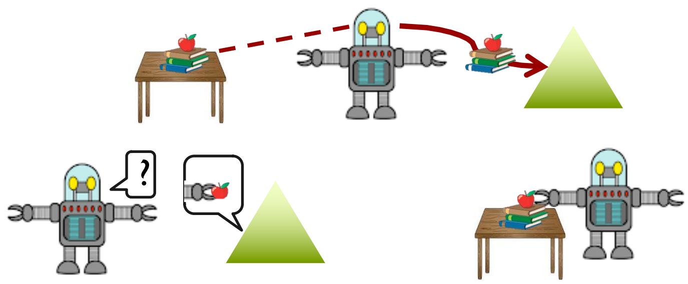
# Knowledge-Based Agents and Propositional Logic (Ch. 7)



*Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer*

1

## Bookkeeping

- If you missed class Tuesday, please collect your exam
  - After today you will need to come to my office for it
- Last class: Information gain; evaluating ML models
- This class: Knowledge-based agents and Propositional logic
- Next class: First order logic
  - If we have time: some common problems with the midterm

2

# Today's Class

- Knowledge Based Agents
  - Knowledge Bases
  - Inference
- The Wumpus
  - A simple case that we can do a lot with using logic
- Inferential Logics
  - Propositional (Boolean) Logic: a Refresher
  - Logic in general - models and entailment
  - Equivalence, validity, satisfiability
  - Inference rules and theorem proving

3

# Logical Agents

- Knowledge-based agents: agents that have an **explicit representation of knowledge** that can be reasoned with
  - We refer to the set of represented knowledge as a "knowledge base"
    - For historical reasons, KBs are shown as triangles
- These agents can manipulate this knowledge to infer new things at the "knowledge level"

Image: www.shutterstock.com/image-vector/cheerful-funny-cartoon-childrens-robot-cute-2407552137

4

# A Knowledge-Based Agent

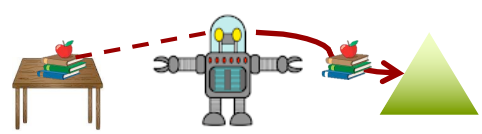
- A knowledge-based agent includes (at least):
  - A **knowledge base**
  - An **inference system**
- Knowledge base (KB) is a set of representations of facts about the world
- Each individual representation is a **sentence** or **assertion**
- Expressed in a **knowledge representation language**
  - Usually starts with some background knowledge
    - Can be general (world knowledge) or specific (domain language)

5

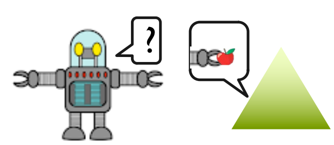
# A Knowledge-Based Agent

- Operates as follows:

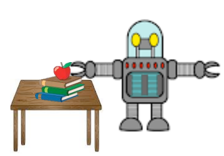
1. **TELLs** the knowledge base what it perceives.
   - ("asserts" knowledge into the KB)



2. **ASKs** the knowledge base what action to perform.
   - (performs "inference")



3. **PERFORMs** the chosen action.



6

# A simple knowledge-based agent must…

- Represent states, actions, etc.
- Incorporate new percepts
- Update internal representations of the world
- Deduce hidden properties of the world
- Deduce appropriate actions

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

   TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
   action ← ASK(KB, MAKE-ACTION-QUERY(t))
   TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   t ← t + 1
   return action
```

7

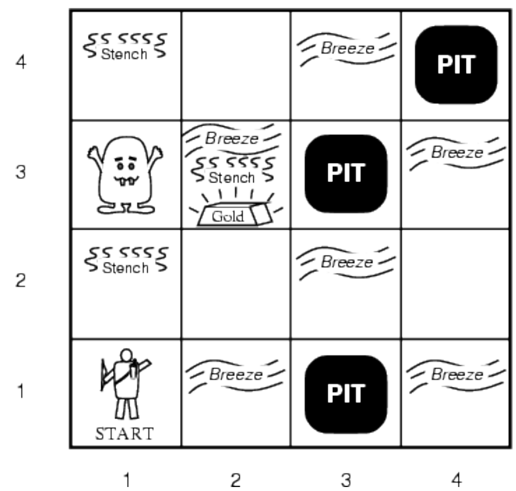
# Architecture of a Knowledge-Based Agent

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

8

## The Wumpus World Environment

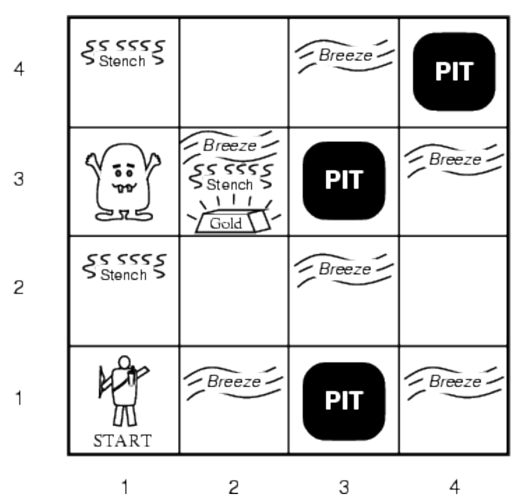
- The Wumpus computer game
  - Agent explores a dark cave consisting of rooms connected by passageways.
  - Lurking somewhere in the cave is the Wumpus, a beast that eats any agent that enters its room.
  - Some rooms contain bottomless pits that trap any agent that wanders into the room.
  - Occasionally, there is a heap of gold in a room.
  - The goal is to collect the gold and exit the world without being eaten (or trapped).




9

## A Typical Wumpus World

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




10

# Wumpus Goal

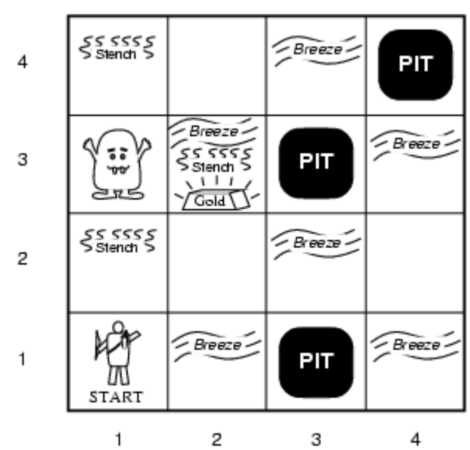
- Agent's goal is to:
  - Find the gold
  - Bring it back to the start square as quickly as possible
  - Don't get killed
- Scoring
  - 1000 points reward for climbing out with the gold
  - 1 point deducted for every action taken
  - 1000 points penalty for getting killed
- Principle Difficulty: agent is initially ignorant of the configuration of the environment – going to have to reason to figure out where the gold is without getting killed!

*slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt*

11

# Wumpus World PEAS description

- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- Environment: 4 x 4 grid of rooms
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square




- Sensors: Stench, Breeze, Glitter, Bump, Scream (shot Wumpus)
- Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot

12

# Wumpus world characterization

- Fully Observable? No – can't see the entire state
- Deterministic? Yes – there's no probability/stochasticity
- Episodic? No – need recollection of past percepts and actions
  - In an episodic environment, only the current percept is required
  - In a sequential environment, an agent requires memory of past actions to determine the next best actions
- Static? Yes – the world doesn't change while the agent thinks
- Discrete? Yes – distinct percepts and actions are discrete
- Single-agent? Yes – the wumpus isn't an agent; it's more like a hazard

13

# Agent in a Wumpus World: Percepts

- Agent perceives
  - **Stench** in the square containing the wumpus and in adjacent squares (not diagonally)
  - **Breeze** in the squares adjacent to a pit
  - **Glitter** in the square where the gold is
  - **Bump**, if it walks into a wall
  - **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 (have to remember over time)

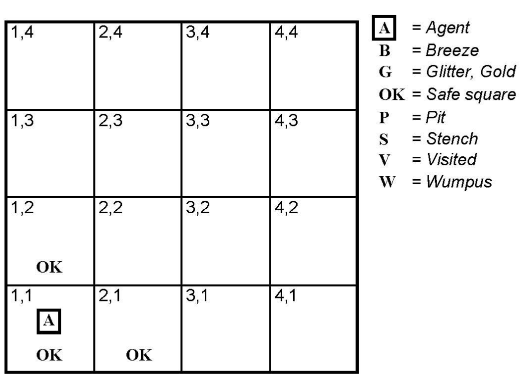
16

## Wumpus Agent Actions

- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab**: Pick up an object that is in the same square as the agent
- **shoot**: Fire an arrow in a straight line in the direction the agent is facing.
  - The arrow continues until it either 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**: leave the cave. This action is only effective in the start square
- **die**: This action automatically happens if the agent enters a square with a pit or a live wumpus

17

## Exploring the Wumpus World




(a)

**Initial situation:**

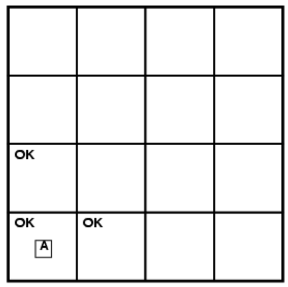
**Agent in 1,1 and percept is [None, None, None, None, None]**

**From this the agent can infer the neighboring squares are safe (otherwise there would be a breeze or a stench)**

*slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt*

20

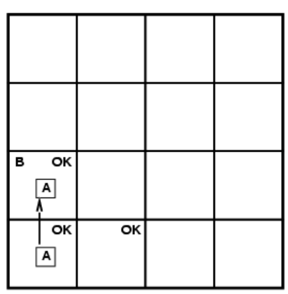
# Exploring a wumpus world






21

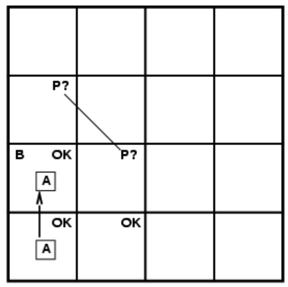
# Exploring a wumpus world






22

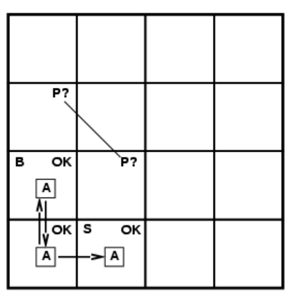
# Exploring a wumpus world




slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

23

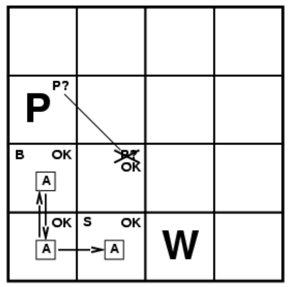
# Exploring a wumpus world




slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

24

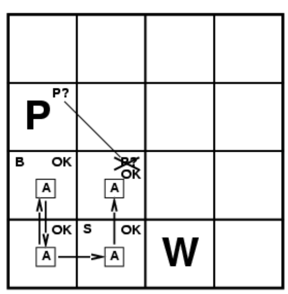
# Exploring a wumpus world




slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

25

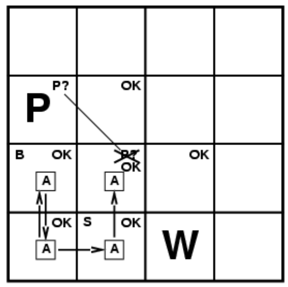
# Exploring a wumpus world




slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

26

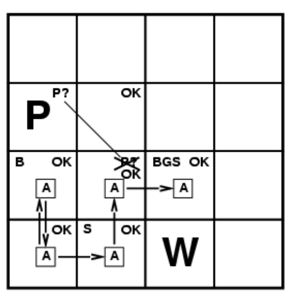
# Exploring a wumpus world




slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

27

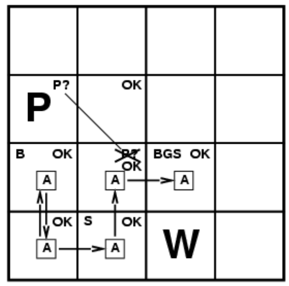
# Exploring a wumpus world




slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

28

# Exploring a wumpus world




**In each case where the agent draws a conclusion from the available information, that conclusion is guaranteed to be correct if the available information is correct…**

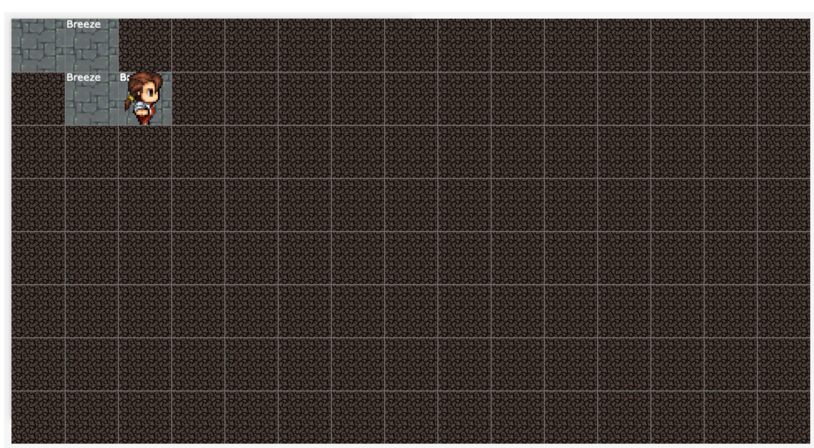
**This is a fundamental property of logical reasoning**

*slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt*

29

# Wumpuses Online

- Google “Hunt the Wumpus” and you will find many playable versions
- There are many variations (multiple gold, bats, multiple arrows, …)



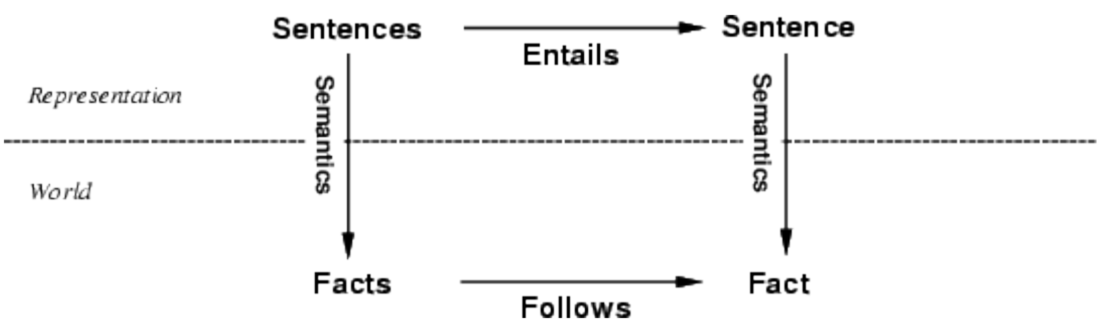

30

## Representation, Reasoning, and Logic

- Point of knowledge representation is to express knowledge in a **computer usable** form
  - Needed for agents to act on it!
- **Logics** are formal languages for representing information such that conclusions can be drawn
- **Syntax** defines how symbols can be put together to form the sentences in the language
- **Semantics** define the "meaning" of sentences;
  - i.e., define truth of a sentence in a world (given an interpretation)
- Knowledge is stored in a Knowledge Base, or KB

32

## 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.

"Dr M is sick with the flu" ⊨ "Dr M is sick"

33

# Entailment

- Entailment means that one thing follows logically from another:
  - KB ⊨ α
  - Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true
  - E.g., the KB containing "the Phillies won and the Astros won" entails "The Phillies won"
- Inference is a procedure that allows new sentences to be derived from a knowledge base.
  - KB ⊢ α
  - E.g., from the KB containing "the Phillies won" and "Dr. M will be happy if the Phillies win," we can infer "Dr. M is happy"

*Some slide material: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt*

34

# Entailment and Derivation

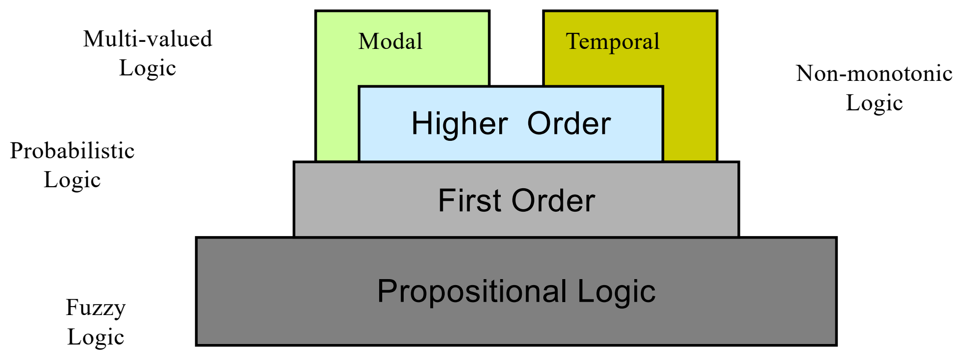
- **Entailment: KB ⊨ Q**
  - Q is entailed by KB (a set of premises or assumptions) if and only if there is no logically possible world in which Q is false while all the premises in KB are true.
  - Or, stated positively, Q is entailed by KB if and only if the conclusion is true in every logically possible world in which all the premises in KB are true.
- **Derivation: KB ⊢ Q**
  - We can derive Q from KB if there is a **proof** consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

x ⊨ y: x semantically entails y

x ⊢ y: y is provable from x

35

# Logic as a KR Language




36

# Ontology and Epistemology

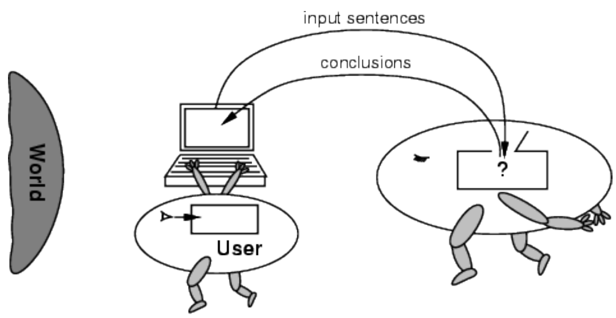
- **Ontology** is the study of what there is—an inventory of what exists. An **ontological commitment** is a commitment to an existence claim.
  - Knowledge bases with certain characteristics are called **ontologies.**
- **Epistemology** is a major branch of philosophy that concerns the forms, nature, 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…1 |
| Fuzzy logic | degree of truth | degree of belief 0…1 |

37

## No Independent World Access

- The reasoning agent often gets its knowledge about the facts of the world as a *sequence of logical sentences.*
- Must draw conclusions from them without (other) access to the world.
- Thus it is very important that the agent's reasoning is sound!




38

## KBs allow reasoning about actions

- Agents have a **knowledge base** that contains everything they know about the world
  - Including goals and possible actions
  - Updated when new percepts arrive from sensors
  - Stored as logical sentences: `(haveColor redColor apples)`
- Inference can be used to work out new facts about the world
  - When new facts come in, can conclude new things
  - Based on that knowledge, make decisions about what to do
- Knowledge underpins decisions about what actions to take

39

## KB Agents - Summary

- Intelligent agents need **knowledge about the world** for making good decisions
- The knowledge of an agent is stored in a knowledge base in the form of **sentences** in a **knowledge representation language**
- A knowledge-based agent needs a **knowledge base** 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 structure of sentences and how they relate to world facts
- The **interpretation** of a sentence is the fact to which it refers. If this fact is part of the actual world, then the sentence is true

40

# Propositional Logic

Chapter 7.4-7.8

41

# Propositional Logic

- **Logical constants:** true, false
- **Propositional symbols**: P, Q, S, ... (atomic sentences)
- **Wrapping parentheses**: ( … )
- **Sentences** are atoms combined by connectives:
  - ∧ ...and [conjunction]
  - ∨ ...or [disjunction]
  - ⇒ ...implies [implication / conditional]
  - ⇔ ...is equivalent [biconditional]
  - ¬ ...not [negation]
- **Literal**: atomic sentence or negated atomic sentence

43

# Propositional Logic (PL)

- A simple language useful for showing key ideas and definitions
- User defines a set of propositional symbols, like P and Q.
- User defines the **semantics** of each propositional symbol:
  - Ho means "It is hot"
  - Hu means "It is humid"
  - R means "It is raining"
- Combinations of propositional symbols yield logical expressions

44

## Propositional logic: Syntax

- Propositional logic is the simplest logic – illustrates basic ideas
  - If S is a sentence, $\neg$(S) is a sentence (negation)
  - If S1 and S2 are sentences, (S1 $\wedge$ S2) is a sentence (conjunction)
  - If S1 and S2 are sentences, (S1 $\vee$ S2) is a sentence (disjunction)
  - If S1 and S2 are sentences, (S1 $\Rightarrow$ S2) is a sentence (implication)
  - If S1 and S2 are sentences, (S1 $\Leftrightarrow$ S2) is a sentence (biconditional)

45

## Propositional logic: Semantics

- Each **model** specifies true/false for each proposition symbol
- Rules for evaluating truth with respect to a set of t/f values for sentences:

| | | | | |
|---|---|---|---|---|
| $\neg$S | is true iff | S is false | | |
| S1 $\wedge$ S2 | is true iff | S1 is true | **and** | S2 is true |
| S1 $\vee$ S2 | is true iff | S1 is true | **or** | S2 is true |
| S1 $\Rightarrow$ S2 | is true iff | S1 is false | **or** | S2 is true |
| S1 $\Leftrightarrow$ S2 | is true iff | S1$\Rightarrow$S2 is true | **and** | S2$\Rightarrow$S1 is true |

- Simple recursive process evaluates an arbitrary sentence, e.g.,
  - $\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1})$ = true $\wedge$ (true $\vee$ false) = true $\wedge$ true = true

*slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt*

46

## Examples of PL Sentences

- (P ∧ Q) ⇒ R : "If it is hot and humid, then it is raining"
- Q ⇒ P : "If it is humid, then it is hot"
- Q : "It is humid."
- A better way:
  - Ho = "It is hot"
  - Hu = "It is humid"
  - R = "It is raining"
- (Ho ∧ Hu) ⇒ R ; Hu ⇒ Ho; Hu

47

## Validity and Satisfiability

- A sentence is valid if it is true in all models
  - Ex: True, A∨¬A, A ⇒ A, (A∧(A ⇒ B)) ⇒ B
- A sentence is satisfiable if it is true in some model
  - Ex: A ∨ B, C
- A sentence is unsatisfiable if it is true in no models
  - Ex: A ∧ ¬A

49

## 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" (assignment of truth values to propositional symbols) in which each sentence in the KB is True.
  - E.g.: it is both hot and humid.

50

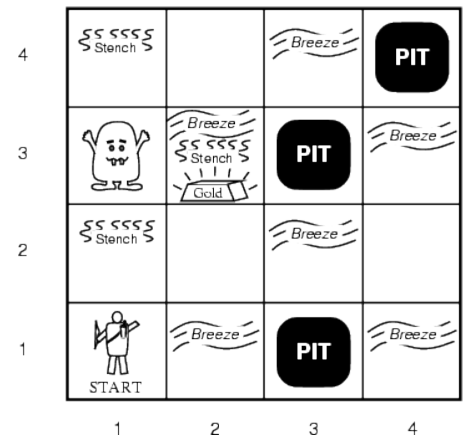
## More Terms

- 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 are.
  - 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.
  - Example: "It's raining and it's not raining."
- P semantically **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.

51

## Wumpus world sentences

- Let $P_{i,j}$ be true if there is a pit in [i, j];
  Let $B_{i,j}$ be true if there is a breeze in [i, j].
  - $\neg P_{1,1}$
  - $\neg B_{1,1}$
  - $B_{2,1}$
- "Pits cause breezes in adjacent squares"
  - $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$




*slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt*

52

## Inference Rules

- **Logical inference** is used to create new sentences that logically follow from a given set of predicate calculus sentences (KB).
- An inference rule is **sound** if every sentence X produced by an inference rule operating on a KB logically follows from the KB.
  - (That is, the inference rule does not create any contradictions)
- An inference rule is **complete** if it is able to produce every expression that logically follows from (is entailed by) the KB.
  - (Note the analogy to complete search algorithms.)

56

## Two Important Properties for Inference

- **Soundness: If KB ⊢ Q then KB ⊨ Q**
  - If Q is derived from a set of sentences KB using a given set of rules of inference, then Q is entailed by KB.
  - Hence, inference produces only real entailments, AKA any sentence that follows deductively from the premises is valid.
- **Completeness: If KB ⊨ Q then KB ⊢ Q**
  - If Q is entailed by a set of sentences KB, then Q can be derived from KB using the rules of inference.
  - Hence, inference produces all entailments, AKA all valid sentences can be proved from the premises.

57

## Sound Rules of Inference

- Here are some examples of sound rules of inference
  - A rule is sound if its conclusion is true whenever the premise is true
- Each can be shown to be sound using a truth table

| RULE | PREMISES | CONCLUSION |
|---|---|---|
| Modus Ponens | $A, A \Rightarrow B$ | $B$ |
| And Introduction | $A, B$ | $A \land B$ |
| And Elimination | $A \land B$ | $A$ |
| Double Negation | $\neg\neg A$ | $A$ |
| Unit Resolution | $A \lor B, \neg B$ | $A$ |
| Resolution | $A \lor B, \neg B \lor C$ | $A \lor C$ |
| de Morgans | $\neg(A \lor B)$ | $\neg A \land \neg B$ |
| $\lor / \Rightarrow$ Equivalence | $A \Rightarrow B$ | $\neg A \lor B$ |

58

## Soundness of Modus Ponens

- If A and A ⇒ B; then B

| A | B | A → B | OK? $(A \land (A \to B)) \to B$ |
|---|---|---|---|
| True | True | True | ✓ |
| True | False | False | ✓ |
| False | True | True | ✓ |
| False | False | True | ✓ |

59

## Proving Things

- A proof is a sequence of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference.
- The last sentence is the theorem (also called goal or query) that we want to prove.
- Example for the "weather problem" given above: **Is it raining (R=true), given Hu?**

| | | |
|---|---|---|
| 1. Hu | Premise | "It is humid" |
| 2. Hu ⇒ Ho | Premise | "If it is humid, it is hot" |
| 3. Ho | Modus Ponens(1,2) | "It is hot" |
| 4. (Ho ∧ Hu) ⇒ R | Premise | "If it's hot & humid, it's raining" |
| 5. Ho ∧ Hu | And Introduction(1,3) | "It is hot and humid" |
| 6. R | Modus Ponens(4,5) | "It is raining" |

61

# Logical equivalence

- Two sentences are logically equivalent iff true in same models: α ≡ ß iff α ╞ β and β ╞ α

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

slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

62

# Horn Sentences

- A Horn sentence or Horn clause has the form:
  P1 ∧ P2 ∧ P3 … ∧ Pn ⇒ Q
- or alternatively
  ¬P1 ∨ ¬P2 ∨ ¬P3 … ∨ ¬Pn ∨ Q

***(P → Q) = (¬P ∨ Q)***

- …where Ps and Q are non-negated atoms
- To get a proof for Horn sentences, apply Modus Ponens repeatedly until nothing can be done
- We will use the Horn clause form later

63

## Propositional Logic is a Weak 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 (abbreviated FOL or FOPC) is expressive enough to concisely represent this kind of information
- FOL adds relations, variables, and quantifiers, e.g.,
  - "Every elephant is gray": $\forall x\ (\text{elephant}(x) \Rightarrow \text{gray}(x))$
  - "There is a white alligator": $\exists x\ (\text{alligator}(X) \wedge \text{white}(X))$

We'll get into FOL next lecture!

64

## 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?

65

# Example II

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

  P = "person"; Q = "mortal"; R = "Confucius"

- so the above 3 sentences are represented as:

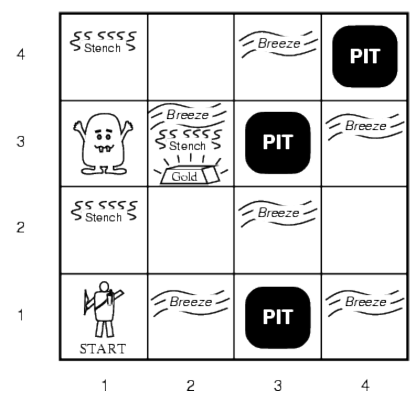
  $P \Rightarrow Q$; $R \Rightarrow P$; $R \Rightarrow Q$

- Although the third sentence is entailed by the first two, we needed an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes "person" and "mortal"
- To represent other individuals we must introduce separate symbols for each one, with some way to represent the fact that all individuals who are "people" are also "mortal"

66

# The "Hunt the Wumpus" Agent

- Some atomic propositions:
  - S12 = There is a stench in cell (1,2)
  - B34 = There is a breeze in cell (3,4)
  - W13 = The Wumpus is in cell (1,3)
  - V11 = We have visited cell (1,1)
  - OK11 = Cell (1,1) is safe.
  - etc




- Some rules:

(Rule 1) $\neg S11 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W21$
(Rule 2) $\neg S21 \rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$
(Rule 3) $\neg S12 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$
(Rule 4) $S12 \rightarrow W13 \vee W12 \vee W22 \vee W11$

- Note that the lack of variables requires us to give similar rules for each cell

67

**YOUR MISSION**

- Prove that the Wumpus is in (1,3) and there is a pit in (3,1), given the observations shown and these rules:

**Rules**

- If there is no stench in a cell, then there is no wumpus in any adjacent cell
- If there is a stench in a cell, then there is a wumpus in some adjacent cell
- If there is no breeze in a cell, then there is no pit in any adjacent cell
- If there is a breeze in a cell, then there is a pit in some adjacent cell
- If a cell has been visited, it has neither a wumpus nor a pit
  - **FIRST** write the propositional rules for the relevant cells
  - **NEXT** write the proof steps and indicate what inference rules you used in each step

# Prove it!

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

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| V12<br>S12<br>-B12 | V22<br>-S22<br>-B22 | | |
| V11<br>-S11<br>-B11 | V21<br>B21<br>-S21 | | |

**INFERENCE RULES**

Modus Ponens
  A, A → B
  ergo B
And Introduction
  A, B
  ergo A ∧ B
And Elimination
  A ∧ B
  ergo A
Double Negation
  ¬¬A
  ergo A
Unit Resolution
  A ∨ B, ¬B
  ergo A
Resolution
  A ∨ B, ¬B ∨ C
  ergo A ∨ C

68

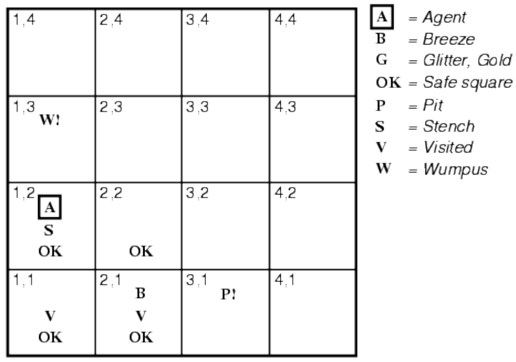
# Proving W13

- Apply MP with ¬S11 and Rule 1:
  ¬ W11 ∧ ¬ W12 ∧ ¬ W21
- Apply And-Elimination to this, yielding three sentences:
  ¬ W11, ¬ W12, ¬ W21
- Apply MP to ~S21 and Rule 2, then apply And-Elimination:
  ¬ W22, ¬ W21, ¬ W31
- Apply MP to S12 and Rule 4 to obtain:
  W13 ∨ W12 ∨ W22 ∨ W11
- Apply Unit Resolution on (W13 ∨ W12 ∨ W22 ∨ W11) and ¬W11:
  W13 ∨ W12 ∨ W22
- Apply Unit Resolution with (W13 ∨ W12 ∨ W22) and ¬W22:
  W13 ∨ W12
- Apply UR with (W13 ∨ W12) and ¬W12:
  W13
- QED

69

# 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




70

# Problems with the Propositional Wumpus Hunter

- 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

71

## Summary: Knowledge-Based Agents

- Knowledge-based agents use a **knowledge base** to store everything they know about the world
  - Everything is represented as **sentences** in that KB
- As they learn more about the world, they make changes to the KB
  - Add and delete facts based on percepts and reasoning
- They use **inference** over represented sentences to perform reasoning
  - Requires the use of an **inference engine**
  - You can draw conclusions from facts and rules by **proving** new sentences
- Simplest kind of logic that we'll see is **propositional logic**
  - It works, but lacks certain capabilities that would be useful

73

## Summary: Inference

- The process of deriving new sentences from old ones is called inference.
  - **Sound** inference processes derives true conclusions given true premises
  - **Complete** inference processes derive all true conclusions from a set of premises
- A **valid sentence** is true in all worlds under all interpretations
- If an implication sentence (rule) 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 regarding the facts
  - Logics are useful for the commitments they do not make because lack of commitment gives the knowledge base engineer more freedom
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - It has a simple syntax and simple semantics. It suffices to illustrate the process of inference
  - Propositional logic quickly becomes impractical, even for very small worlds

74