# Artificial Intelligence
# CS 6364

Professor Dan Moldovan

## Section 4

## Logical Agents

# Knowledge-Based Agents

- Knowledge-Based Agents
- An Example:  Wumpus
- K. Representation, and Reasoning
- Propositional Calculus
- Reasoning about the Wumpus world

<u>Knowledge-Based Agents</u>

The two most important components of AI systems are: Knowledge base and reasoning

A KB is a set of representations of facts; representations are called sentences. Unlike in DB, the KB representation is such that allows reasoning.

The key issues that need to be addressed are:  how to represent the knowledge, and to find inference rules that allow us to reason on the KB.

The agent perceives the outside world and puts more facts on the KB, and then makes decisions about future actions.

# Knowledge-Based Agents

One can distinguish several knowledge levels:

| Level | Primitives |
|---|---|
| Epistemological level | Concept types, inheritance and structuring relations |
| Logical level | Propositions, predicates, logical operators |
| Implementation level | Atoms, pointers, data structures |

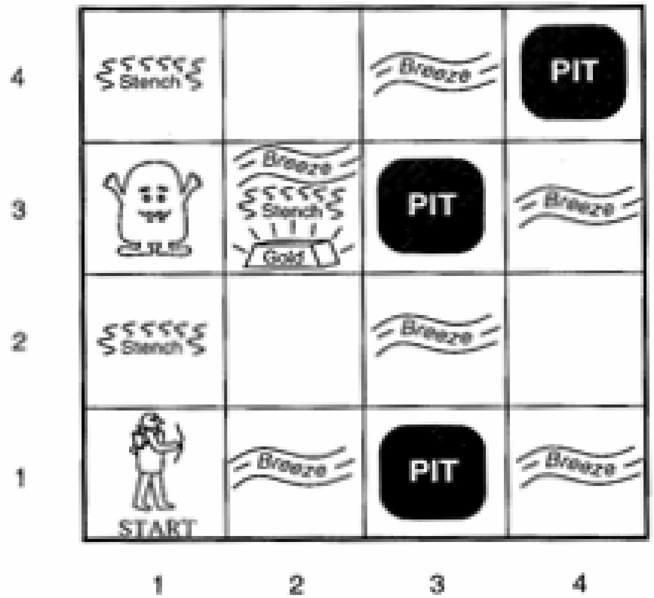
Example:

Epistemological level:  Golden Gate Bridge links San Francisco and Marin County.

Logical level:  Links (GG Bridge, SF, Marin)

Implementation level:  pick up a data representation for above.

# The Wumpus Example




<u>Percepts</u>:

- A stench in adjacent squares and in the square containing wumpus
- A breeze in squares adjacent to a pit
- A glitter in squares with gold
- A bump when agent goes into a wall
- A scream when wumpus is killed

Form a reading vector

(Stench, Breeze, Glitter, Bump, Scream)

with values 1 or 0 as percepts indicate.

# 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

# The Wumpus Example

Actions

- Go forward, turn right 90°, turn left 90°, grab, shoot, climb,
- Agent dies, Wumpus dies.

Goal

- 1000 points for getting gold out of cave, 1 point penalty for each action taken, 10,000 point penalty for getting killed.

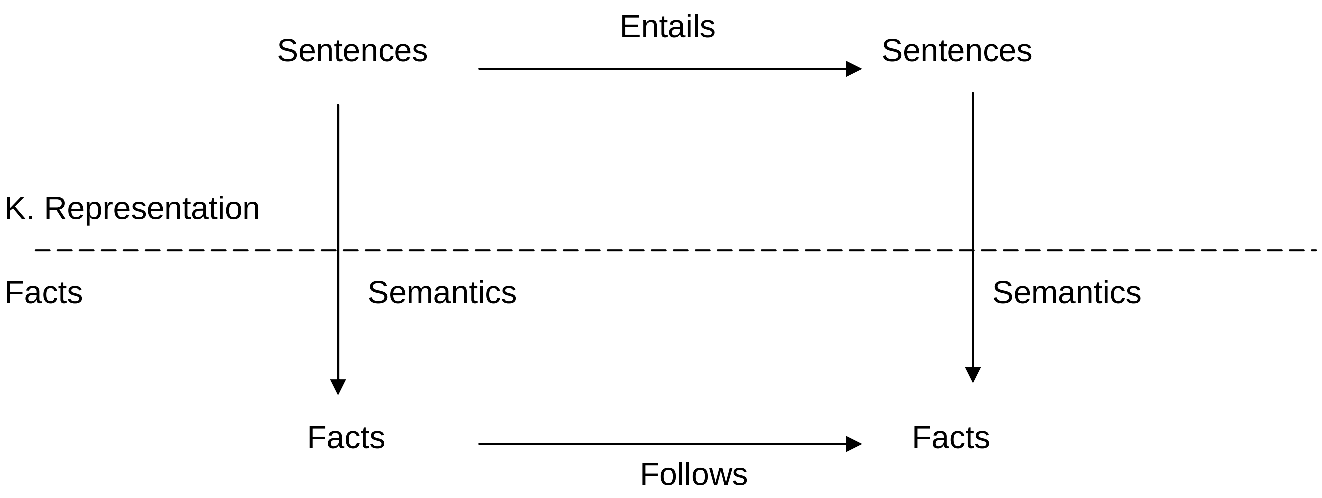
The game repeats several times with different initial positions.

# K. Representation and Reasoning

A K.R. language, like any language, has:

句法
- Syntax—specifies possible forms that sentences can take
- Semantics—determines the facts in the world to which sentence refers.

语义



# Possible worlds or models

A sentence has to be considered with respect to a possible world or model.

Ex. $x + y = 4$ is true when $x = 2$ and $y = 2$
is false when $x = 1$ and $y = 1$

Ex. Clinton is the US President.
was true in the worlds during Jan. 1993 – Jan. 2001 but is not true today.

Ex. Moldovan is at UTD.
various interpretations are possible

Note – Possible worlds are related to contexts.

If a sentence $\alpha$ is true in a model m we say that $\alpha$ satisfies m, or m is a model of $\alpha$.

M ($\alpha$) – defines the set of all models of $\alpha$

# Entailment

Logical reasoning is based on the concept of entailment -
a sentence follows logically from another sentence.

$\alpha \models \beta$ sentence $\alpha$ entails $\beta$

Definition:

$\alpha \models \beta$ if and only if, in every model in which $\alpha$ is true, $\beta$ is also true,

Mathematically:

$\alpha \models \beta$ if and only if $M(\alpha) \subseteq M(\beta)$

$\alpha$ is stronger (more specific) than $\beta$

$M(\alpha)$ are fewer than $M(\beta)$

Entailment is a relationship between sentences (syntax) that is based on semantics

# K. Representation and Reasoning

KB $\vDash \alpha$ "KB entails $\alpha$".

Sentences $\alpha$ may be inferred from KB using inference procedures.

KB $\vdash_i \alpha$ "$\alpha$ is derived from KB using inference procedure i".

Entailment generates only true sentences given that the KB contains true sentences.

# K. Representation and Reasoning

Inferences may be:

- Sound—when generates only entailed sentences—that are true. (truth-preserving).

  Proof—is the procedure of getting sound inference (the steps toward the inference).
- Complete—if it can find a proof for any sentence that is entailed from a KB.

# K. Representation Languages

- Programming languages (C, Pascal, Lisp) lack the expressiveness and are not adequate for KR.
  - Ex:  All men are mortal.
- Natural languages—are expressive, but since have evolved as a way of communication, they have ambiguities and use expressions that are very difficult to encode.
  - The solution for a good KR may be leaning more towards NL (personal opinion)

# Propositional Logic

We look at: syntax, semantics, inference rules

Syntax

- Symbols representing propositions:

  P,Q, R,...

  Ex: P: John is bold.
- Logical constants: True, False
- Wrapping parentheses ( ) that group symbols
- Connectives

  $\land$, $\lor$, $\Rightarrow$, $\rightarrow$, $\Leftrightarrow$, $\neg$

Precedence of connectives in propositional logic

$\neg$, $\land$, $\lor$, $\Rightarrow$ and $\Leftrightarrow$

$\neg P \lor Q \land R \Rightarrow S$ is equivalent to $((\neg P)\lor(Q\land R)) \Rightarrow S$

# Syntax

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

**Figure 7.7** A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

# Propositional Logic

Semantics results from the meaning of proposition symbols, constants and logical connectives.

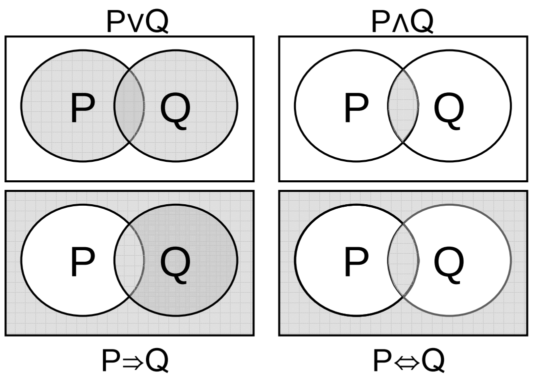
Sentence is:

- Valid if is true for all interpretations
- Satisfiable if is true for some interpretations
- Unsatisfiable if is false for all interpretations

北
归

| *P* | *Q* | *¬P* | *P∧Q* | *P∨Q* | *P⇒Q* | *P⇔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* |

Figure 7.8 Truth tables for the five logical connectives.




Models of complex sentences in terms of the models of their components. In each diagram, the shaded parts correspond to the models of the complex sentences.

# Propositional Logic

Some useful equivalent expressions

| | | | |
|---|---|---|---|
| $P \wedge (Q \wedge R)$ | $\Leftrightarrow$ | $(P \wedge Q) \wedge R$Associativity of conjunction | |
| $P \vee (Q \vee R)$ | $\Leftrightarrow$ | $(P \vee Q) \vee R$Associativity of disjunction | |
| $P \wedge Q$ | $\Leftrightarrow$ | $Q \wedge P$ | Commutativity of conjunction |
| $P \vee Q$ | $\Leftrightarrow$ | $Q \vee P$ | Commutativity of disjunction |
| $P \wedge (Q \vee R)$ | $\Leftrightarrow$ | $(P \wedge Q) \vee (P \wedge R)$ | Distributivity of $\wedge$ over $\vee$ |
| $P \vee (Q \wedge R)$ | $\Leftrightarrow$ | $(P \vee Q) \wedge (P \vee R)$ | Distributivity of $\vee$ over $\wedge$ |
| $\neg(P \wedge Q)$ | $\Leftrightarrow$ | $\neg P \vee \neg Q$ | de Morgan's Law |
| $\neg(P \vee Q)$ | $\Leftrightarrow$ | $\neg P \wedge \neg Q$ | de Morgan's Law |
| $P \Rightarrow Q$ | $\Leftrightarrow$ | $\neg Q \Rightarrow \neg P$ | Contraposition |
| $\neg \neg P$ | $\Leftrightarrow$ | $P$ | Double Negation |
| $P \Rightarrow Q$ | $\Leftrightarrow$ | $\neg P \vee Q$ | |
| $P \Leftrightarrow Q$ | $\Leftrightarrow$ | $\neg P \vee Q$ | |
| $P \Leftrightarrow Q$ | $\Leftrightarrow$ | $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$ | |
| $P \Leftrightarrow Q$ | $\Leftrightarrow$ | $(P \wedge Q) \vee (\neg P \wedge \neg Q)$ | |
| $P \wedge \neg P$ | $\Leftrightarrow$ | False | |
| $P \vee \neg P$ | $\Leftrightarrow$ | True | |

# Propositional Logic

Truth tables may be used for proving the validity of small sentences.

$$((P \vee H) \wedge \neg H \Rightarrow P$$

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

Truth table showing validity of a complex sentence.

A proposition with $n$ symbols requires $2^n$ rows of truth table, thus is impractical.

# Inference Rules for Propositional logic

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

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

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

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

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

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

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

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

# Inference Rules for Propositional logic

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

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

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

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

- **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.)

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

Seven inference rules for propositional logic. The unit resolution rule is a special case of the resolution rule, which in turn is a special case of the full resolution rule for first-order logic.

# An Agent for the Wumpus World

Sketch of a solution.

Idea: Agent starts with some initial KB. Through percepts, it gathers more facts. Then, using some rules, it concludes about where Wumpus may and may not be. Finally, either by truth table—or (better) using inference rules, narrows down the exact position of Wumpus.

Initial KB informs agent where Wumpus and pits are in respect with percepts received. (We will show this after the percepts).

Say that at a moment the percepts ( i.e. KB ) are

| | |
|---|---|
| $\neg S_{1,1}$ | $\neg B_{1,1}$ |
| $\neg S_{2,1}$ | $B_{2,1}$ |
| $S_{1,2}$ | $\neg B_{1,2}$ |

# An Agent for the Wumpus World

The rules stored in the initial KB are:

**R**: if there is stench in $S_{i,j}$ then W may be in either (i +1, j); (i -1,j); (i, j -1); (i, j +1); (i, j).

**R′**: if there is no stench in $S_{i,j}$ then no W in (i +1, j) and (i -1, j) and (i, j -1), (i, j +1) and (i, j).

For the percept received, R′ translates into:

$R_1$: $\neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$

$R_2$: $\neg S_{2,1} \Rightarrow \neg W_{1,2} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$

Similarly R translates into:

$R_3$: $S_{1,2} \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$

Use truth table and find that

$KB \Rightarrow W_{1,3}$

The table is large, so we use inference rules.

# An Agent for the Wumpus World

1. Modus Ponens with $\neg S_{1,1}$ and $R_1$

   $\neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$
2. And-Elimination

   $\neg W_{1,1}$ , $\neg W_{1,2}$ , $\neg W_{2,1}$
3. Modus Ponens with $\neg S_{2,1}$ and $R_2$

   Plus And-Elimination

   $\neg W_{2,2}$ $\neg W_{2,1}$ $\neg W_{3,1}$
4. Modus Ponens to $S_{1,2}$ and $R_4$

   $W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$
5. Unit Resolution between 4 and 2

   $W_{1,3} \vee W_{2,2} \vee W_{1,1}$

# An Agent for the Wumpus World

6. Unit Resolution 5 and 2

   $W_{2,2} \vee W_{1,3}$

7. Unit Resolution 6 and 3

   $W_{1,3}$

Note that we have freedom in combining sentences available in KB the way we want, and also we may pick any inference rule we want. The problem is that machine cannot figure this easily.