Title: Logical Agents
AIMA: Chapter 6

Corrected version

Introduction to Artificial Intelligence
CSCE 476-876, Spring 2000
`www.cse.unl.edu/~cse476`

Berthe Y. Choueiry
`choueiry@cse.unl.edu, (402)472-5444`

Outline

- Knowledge bases
- Wumpus world
- Logic in general
- Propositional (Boolean) logic
- Inference rules
- Normal forms

Section 6.5 left out as an exercise

Knowledge Base

A fact in the world :

A representation of a fact in the world

A sentence= a representation of a fact in the world in a <u>formal language</u>

A Knowledge Based (KB) :

A set (of representations) of facts about the world

Access to KB :

Two primitives: (1) ASK and (2) TELL

Level of Knowledge

Agents can be viewed at various levels:

1. **Epistemological**:
   Abstract description of what the agent knows about the world

2. **Logical**:
   Encoding of knowledge into sentences

3. **Implementation**:
   Actual implementation (lists, arrays, hash tables, etc.)
   - Very important for performance of agent
   - Irrelevant for higher levels of knowledge

On Knowledge

Declarative knowledge :

The designer tells the agent what it should know about the world, fact by fact

Learning mechanisms :

The agent can generate general knowledge about the world given a series of percepts

Learning in a KB-Agent :

$\rightarrow$ allow building fully autonomous agents

Knowledge-Based Agent (I)

Two components:

1. **Knowledge base**:
   Stores the representations of facts of the world
   May contain some background knowledge

2. **Inference mechanism**:
   Generates new facts from the ones stored in KB

Knowledge-Based Agent (II)

Perceives

$\rightarrow$ Tells KB about new percepts (new sentences)

Reasons

$\rightarrow$ Asks KB about actions to take (inference)

Acts

$\rightarrow$ Tells KB about actions (new sentences)

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

The agent must be able to:

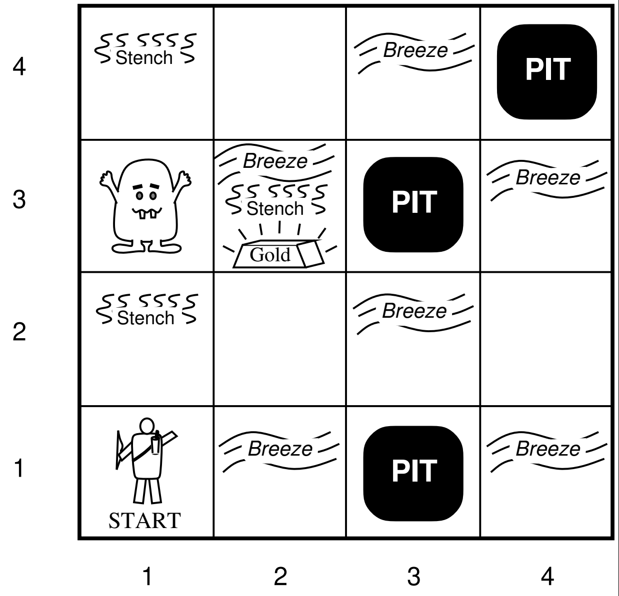
- represent states, actions, etc.
- incorporate new percepts
- update internal representations of the world
- deduce hidden properties of the world
- deduce appropriate actions

Motivating example: The Wumpus world

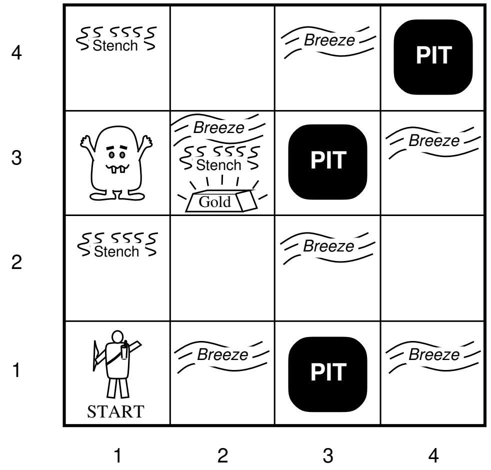
Early computer game

Agent explores a cave with:

- bottomless pits
- a beast that eats anyone who enters the room, and
- heap of gold to trap



Wumpus World (I) PAGE description



Percepts : Breeze, Glitter, Smell, Bump, Wumpus' scream

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

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

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

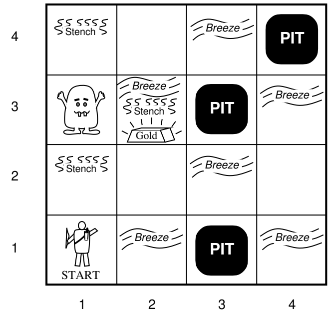
Grabbing picks up the gold if in the same square

Releasing drops the gold in the same square

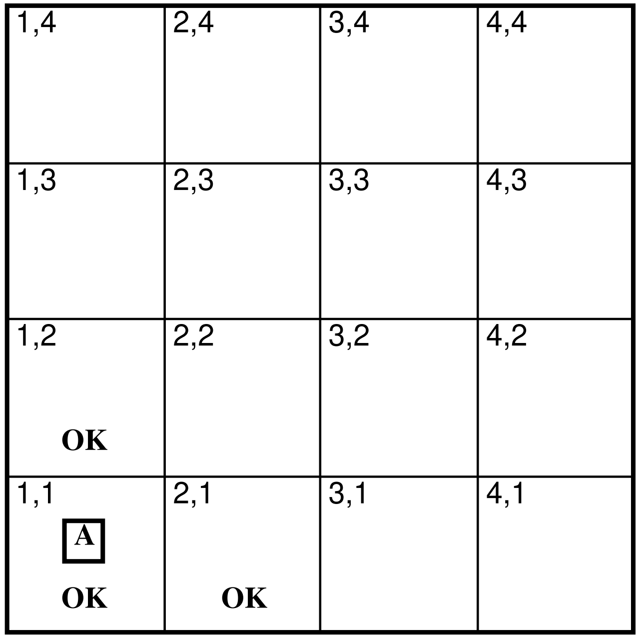
Wumpus World (II) Characterization

Is the world:

- Deterministic? Yes, outcome exactly specified
- Fully accessible? No, only local perception
- Static? Yes, Wumpus/Pits don't move
- Discrete? Yes

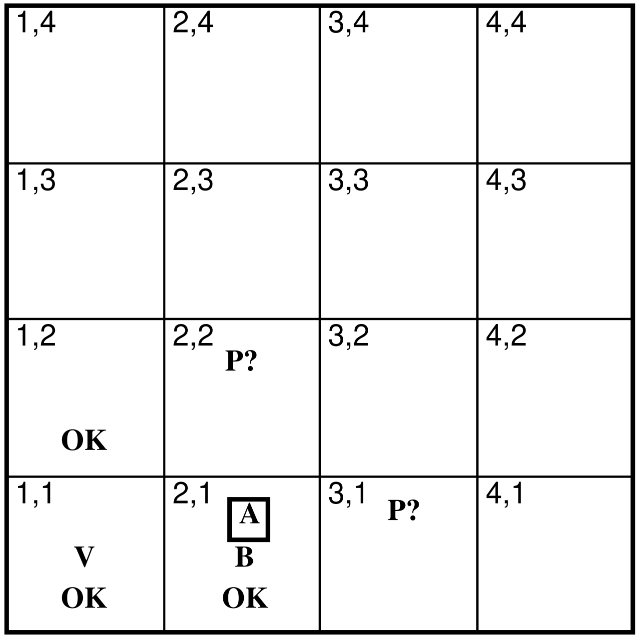


Wumpus World (III) Acting & Reasoning

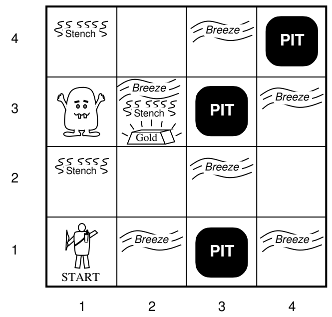


(a)

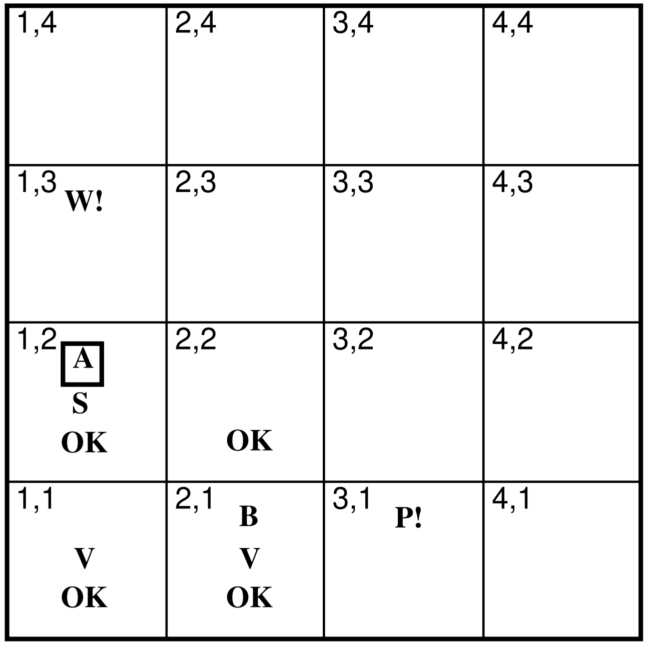
- **A** = *Agent*
- **B** = *Breeze*
- **G** = *Glitter, Gold*
- **OK** = *Safe square*
- **P** = *Pit*
- **S** = *Stench*
- **V** = *Visited*
- **W** = *Wumpus*



(b)

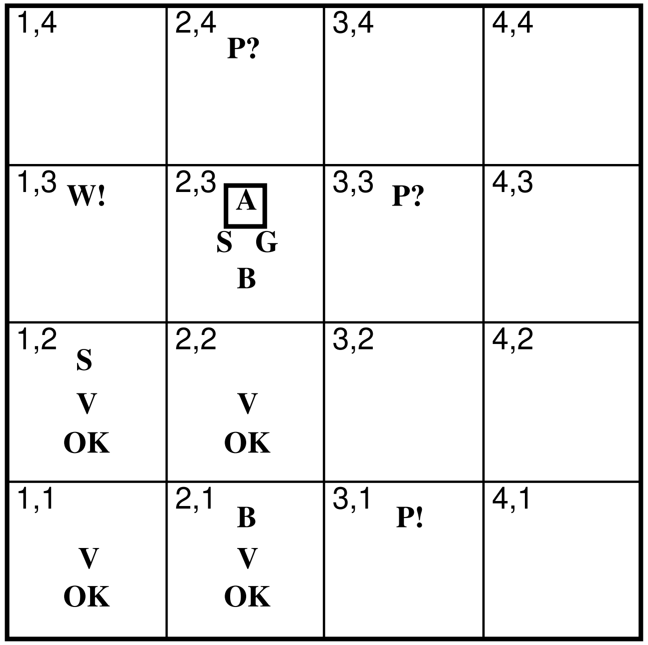


Wumpus World (III) Acting & Reasoning



(a)

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



(b)

Logic in general

Logics are formal languages for representing information such that conclusions can be drawn

Syntax defines the sentences in the language (grammar)

Semantics define the "meaning" of sentences; *i.e.*, define <u>truth</u> of a sentence in a world

Examples, the language of arithmetic:

- Syntax: $x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence
- Semantics:
  – True: $x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y
  – True: $x + 2 \geq y$ is true in a world where $x = 7,\ y = 1$
  – False: $x + 2 \geq y$ is false in a world where $x = 0,\ y = 6$

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 (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

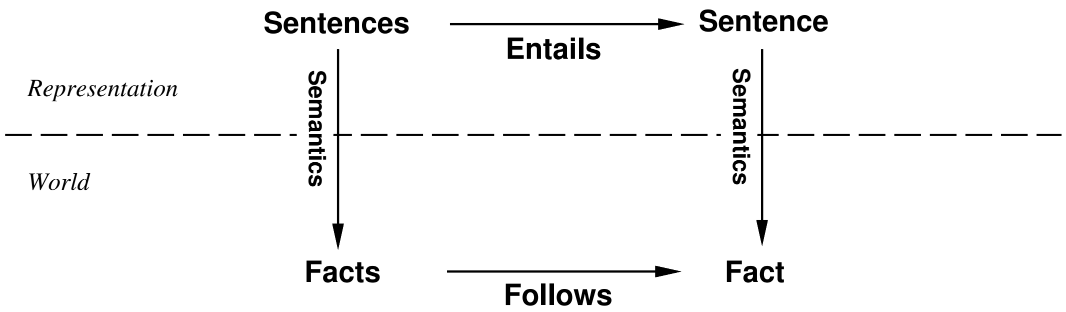
Entailment

Facts : in the world

Representations : in the computer

Reasoning : process of constructing new representations from old ones

Proper Reasoning : ensures new representations correspond to facts that actually follow from facts in the world



Knowledge base KB <u>entails</u> sentence α (KB $\models \alpha$)

iff α is true in all worlds where KB is true

Example: KB: $\{a \wedge b\}$, then KB $\models a$; KB $\models b$; KB $\models a \vee b$;

Inference

A inference proc. can be described by the sentences it can derive

$$\text{KB} \vdash_i \alpha \equiv \alpha \text{ is derived from KB by procedure } i$$

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 $\vdash_i \alpha$

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

The record of operation of a sound inference procedure is a **proof**

A logic consists of:

1. A formal representation system:
   (a) Syntax: how to make sentences
   (b) Semantics: systematics constraints on how sentences relate to the states of affairs
2. Proof theory: a set of rules for deducing the entailment of a set of sentences

Example:

√ Propositional logic (or Boolean logic)

√ First-order logic FOL

Propositional logic: Syntax

Propositional logic is the simplest logic—illustrates basic ideas

- Symbols represent whole propositions, sentences
  D says the Wumpus is dead
  The proposition symbols P_1, P_2, etc are sentences

- Boolean connectives: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$, connect sentences
  If S_1 and S_2 are sentences, the following are sentences too:
  $\neg S_1$, $\neg S_2$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$

Note:

— Premise, antecedent, implication, conclusion, consequence, etc.

— Atomic sentence, literal, complex sentence

— Precedence order (highest to lowest): $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

  ($e.g.$, $\neg P \vee Q \wedge R \Rightarrow S$?)

First-order logic: Syntax (Chapter 7)

First-Order Logic (FOL) is expressive enough to say almost anything of interest and has a sound and complete inference procedure

- Logical symbols:
  - parentheses
  - connectives ($\neg, \Rightarrow$, the rest can be regenerated)
  - variables
  - equality symbol (optional)
- Parameters:
  - quantifier $\forall$
  - predicate symbols
  - constant symbols
  - function symbols

Propositional logic: Semantics

Semantics is defined by specifying:

— Interpretation of a proposition symbols and constants (T/F)

— Meaning of logical connectives

Proposition symbol means what ever you want:

D says the Wumpus is dead

Breeze says the agent is feeling a breeze

Stench says the agent is perceiving an unpleasant smell

Connectives are functions: complex sentences meaning derived from the meaning of its parts

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*

Note:

$P \Rightarrow Q$: if P is true, Q is true, otherwise I am making no claim

Models (I)

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

We say m is a <u>model</u> of a sentence α if α is true in m

In propositional logic:

- A model is a mapping from proposition symbols directly to truth or falsehood
- The models of a sentence are the mappings that make the sentence true

Example:

	α:	obj1 $\wedge$ obj2
$\surd$	Model1:	$1 \wedge 1$
$\surd$	Model2:	$1 \vee 1$
$\times$	Model3:	$0 \wedge 1$

Models (II)

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

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

KB $\models \alpha$ iff all models of KB are models of α (*i.e.*, $M(KB) \subseteq M(\alpha)$)

Propositional logic: 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—α must be true wherever KB is true

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
F	F	F				
F	F	T				
F	T	F				
F	T	T				
T	F	F				
T	F	T				
T	T	F				
T	T	T				

Validity and inference

A sentence is valid if it is true in all models

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

To establish validity, use truth tables:

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*

Validity is connected to inference via the Deduction Theorem:

KB $\models \alpha$ iff (KB $\Rightarrow \alpha$) is valid

Satisfiability and inference

A sentence is satisfiable if it is true in some model
e.g., $A \vee B$, C

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 $\models \alpha$ iff (KB $\wedge \neg\alpha$) is unsatisfiable
i.e., prove α by *reductio ad absurdum*

Proof methods

Proof methods divide into (roughly) two kinds:

Model checking

Truth-table enumeration (sound and complete for propositional)

Heuristic search in model space (sound but incomplete)

e.g., the GSAT algorithm (Exercise AIMA 6.15)

Application of inference rules

Legitimate (sound) generation of new sentences from old

<u>Proof</u> = a sequence of inference rule applications, can use inference rules as operators in a standard search algorithm

Inference rules for Propositional logic (I)

- Modus Ponens (Implication-Elimination)

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

- And-Elimination

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

- And-Introduction

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

- Or-Introduction

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

Inference rules for Propositional logic (II)

- Double-Negation Elimination

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

- Unit resolution

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

- Resolution: $\frac{\alpha\vee\beta, \quad \neg\beta\vee\gamma}{\alpha\vee\gamma}$ or equivalently $\frac{\neg\alpha\Rightarrow\beta, \quad \beta\Rightarrow\gamma}{\neg\alpha\Rightarrow\gamma}$

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
False	*True*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*	*True*
True	*False*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*False*	*True*
True	*True*	*True*	*True*	*True*	*True*

The job of an inference procedure is

to construct proofs

by finding appropriate sequences of applications of inference rules

starting with sentences initially in KB

and culminating in the generation of the sentence
whose proof is desired

Complexity of propositional inference

Truth-table : Complete. 2^n rows: exponential, thus impractical

Inference rules : NP-complete in general

Two special cases:

1. Horn sentences: $P_1 \wedge P_2 \wedge \ldots P_n \Rightarrow Q$, where P_i and Q are non-negated atoms
   Apply Modus Ponens repeatedly: complete and polynomial time.

2. Use locality and exploit monotonicity
   Locality: Modus Ponens
   Monotonicity: if $\mathrm{KB}_1 \models \alpha \Rightarrow KB_1 \cup KB_2 \models \alpha$

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

Conjunctive Normal Form (CNF)

$\textit{conjunction}$ of $\underbrace{\textit{disjunctions} \text{ of } \textit{literals}}_{\textit{clauses}}$

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

Disjunctive Normal Form (DNF)

$\textit{disjunction}$ of $\underbrace{\textit{conjunctions} \text{ of } \textit{literals}}_{\textit{terms}}$

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

Horn Form

conjunction of *Horn clauses* (clauses with ≤ 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 $(C \wedge D) \Rightarrow B$

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

Truth-table method is sound and complete for propositional logic

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

Propositional logic suffices for some of these tasks

List of terms for glossary:

Due date: Mon, Feb 21, 2000

Of course, include words in margins in AIMA

* = Please check books/dictionnaries/resources other than AIMA

- Conjunctive Normal Form (CNF) *
- Disjunctive Normal Form (DNF) *
- Horn clauses *
- Horn Normal Form *
- Models (in logic. Sorry, might be difficult)
- Propositional logic *
- Truth-preserving (inference procedure)