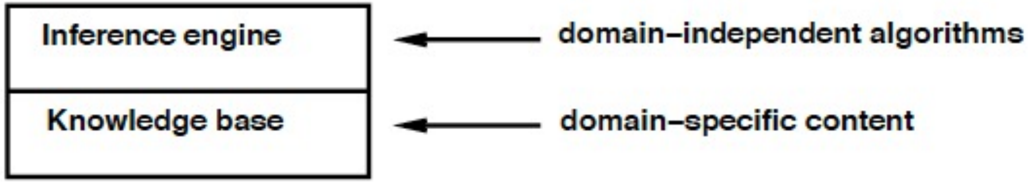
CS 482/628 – AI: Logical Agents

Fall 2021

Knowledge Bases



Knowledge base = set of sentences in a formal language

Declarative approach to building an agent (or other system):

TELL it what it needs to know

Then it can ASK itself what to do—answers should follow from the KB

Agents can be viewed at the knowledge level

i.e., what they know, regardless of how implemented

Or at the implementation level

i.e., data structures in KB and algorithms that manipulate them

A Simple Knowledge-based Agent

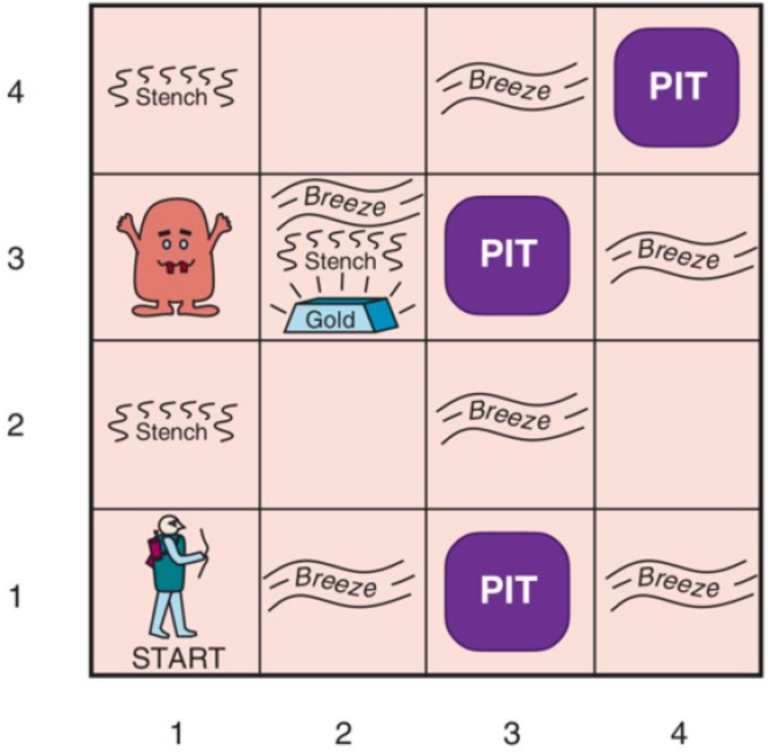
function KB-AGENT(*percept*) **returns** an *action*
  static: *KB*, a knowledge base
    t, a counter, initially 0, indicating time

  TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))
  action ← ASK(*KB*, MAKE-ACTION-QUERY(*t*))
  TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))
  $t \leftarrow t + 1$
  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

The Wumpus World



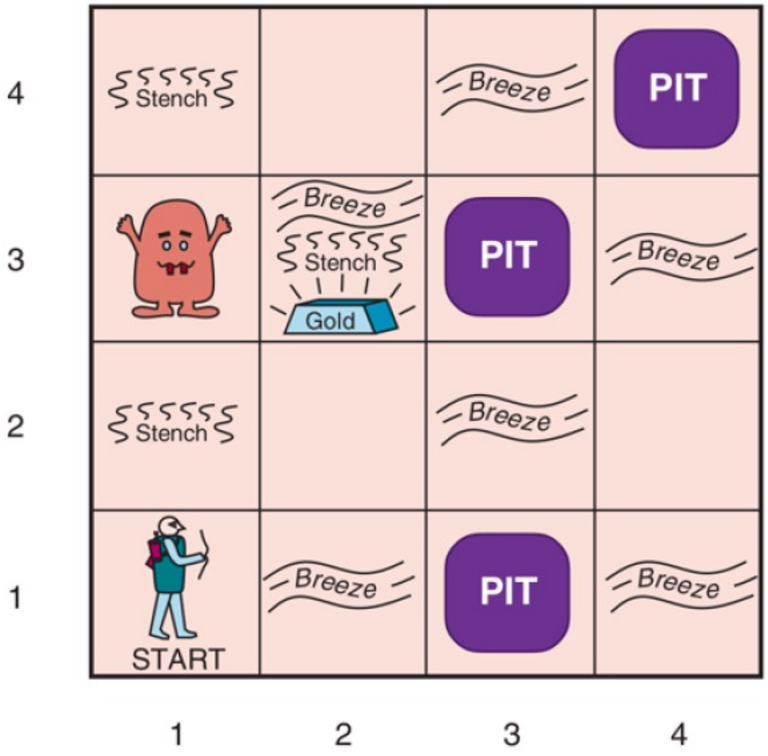
Grid-like world:

- noble hero
- horrible wumpus
- bottomless pits
- gold

Percepts:

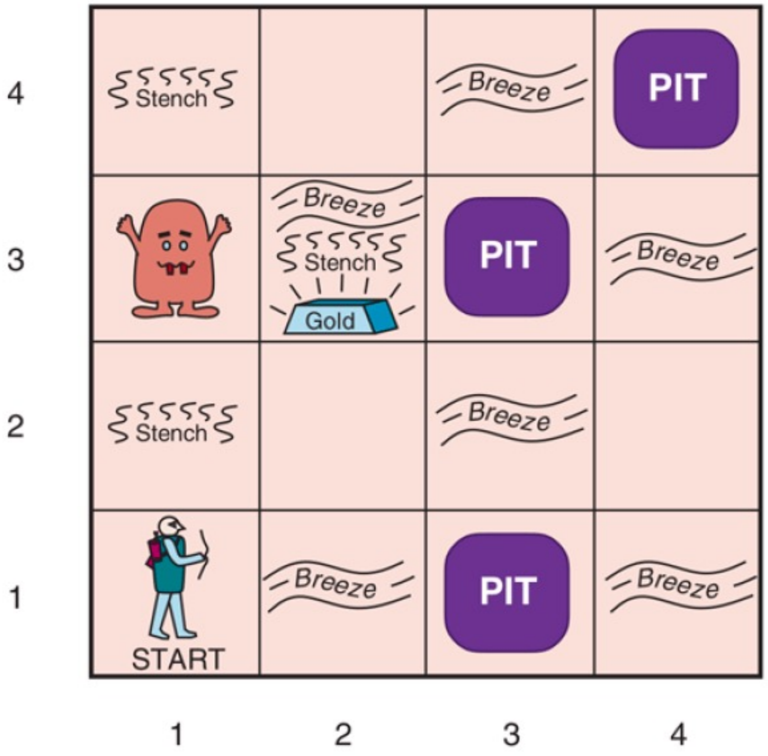
- breeze
- stench
- bump
- glitter
- scream

Actions in the World



- Goals
  - find the gold
  - kill the wumpus
  - go home
- Actions
  - move (N, S, E, W)
  - grab (gold)
  - shoot (N, S, E, W; only one arrow)

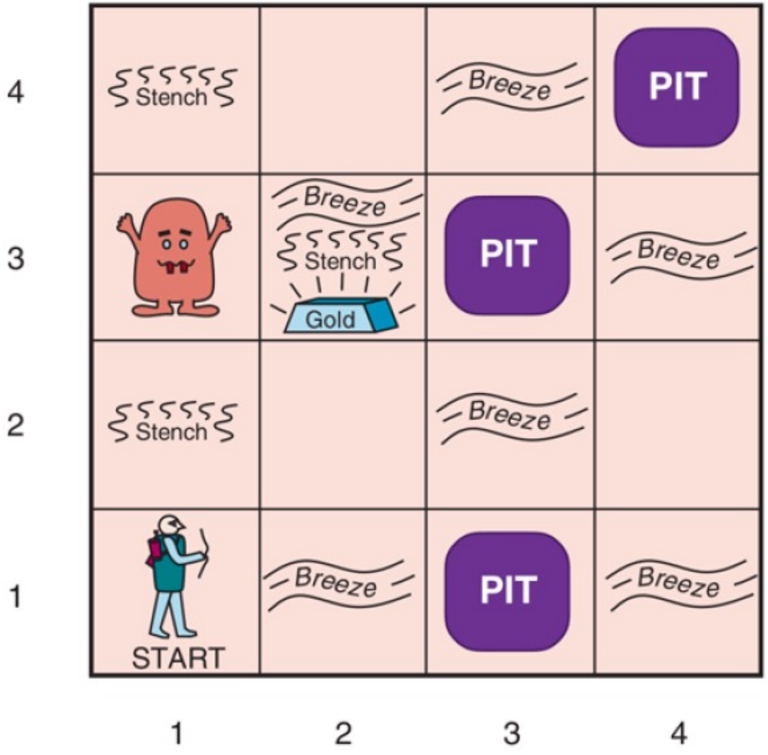
The Wumpus World



If we had complete knowledge of the world, then we could simply build a search tree

What if our perceptions are limited?

Incomplete Knowledge of the World



Agent's percepts:

- stench
- breeze
- glitter
- bump
- scream

Other than the agent, the world is static

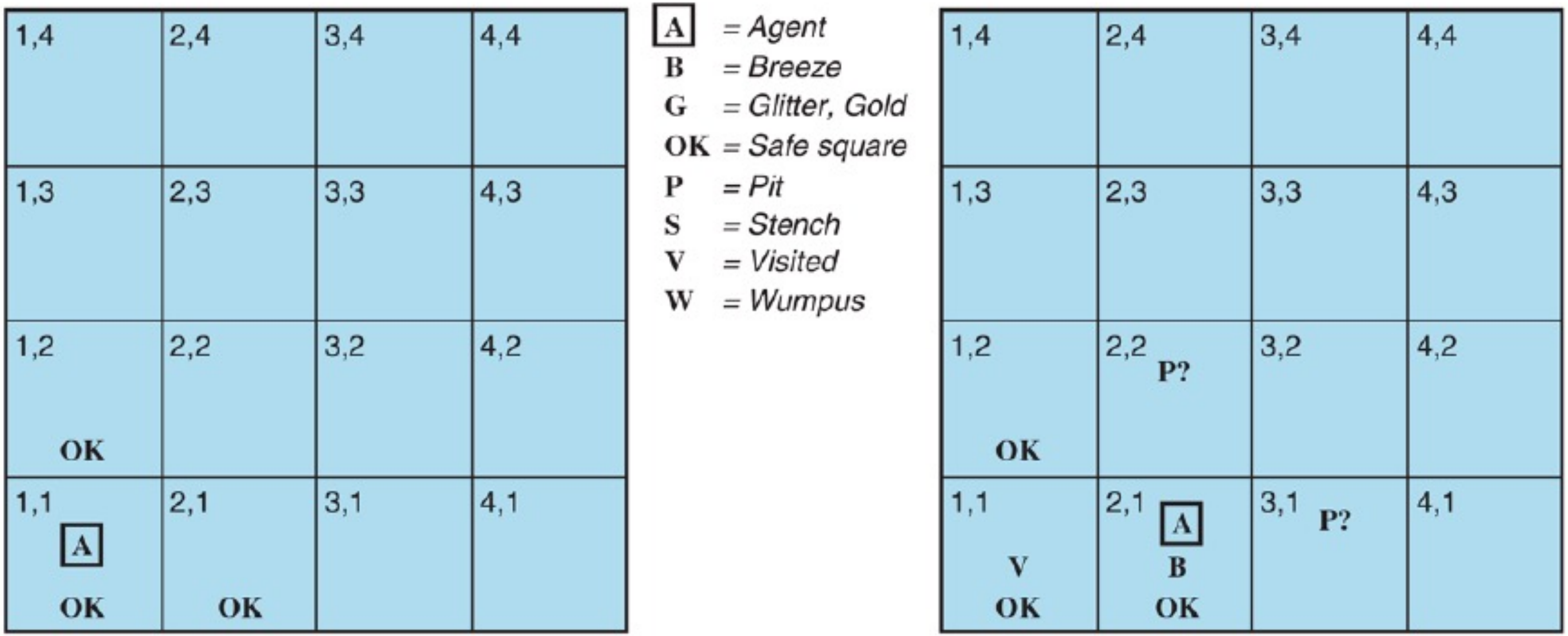
Wumpus World Characterization

Is the world deterministic?? Yes—outcomes exactly specified

Is the world fully accessible?? No—only local perception

Is the world static?? Yes—Wumpus and Pits do not move

Is the world discrete?? Yes

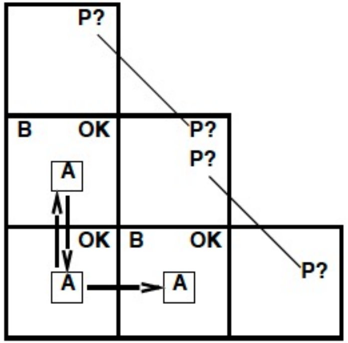


(a) (b)

The first step taken by the agent in the wumpus world. (a) The initial situation, after percept [*None*,*None*,*None*,*None*,*None*]. (b) After moving to [2,1] and perceiving [*None*,*Breeze*,*None*,*None*,*None*].

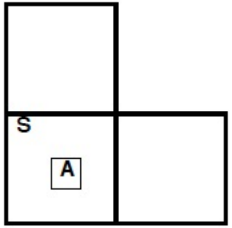
Two later stages in the progress of the agent. (a) After moving to [1,1] and then [1,2], and perceiving [*Stench,None,None,None,None*]. (b) After moving to [2,2] and then [2,3], and perceiving [*Stench,Breeze,Glitter,None,None*].

Other tight spots



Breeze in (1,2) and (2,1)
$\Rightarrow$ no safe actions

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



Smell in (1,1)
$\Rightarrow$ cannot move

Can use a strategy of <u>coercion</u>:
shoot straight ahead
wumpus was there $\Rightarrow$ dead $\Rightarrow$ safe
wumpus wasn't there $\Rightarrow$ safe

Representing Beliefs

In most programming languages, it is easy to specify statements like this...

- *There is a pit in square [3,1]*

But it is difficult to specify statements like these...

- *There is a pit in either square [3,1] or [2,2]*
- *There is no wumpus in square [2,2]*
- *Because there was no breeze in square [1,2], there is a pit in square [3,1]*

Require an agent that can represent this knowledge and perform the reasoning to infer new conclusions

Components of a Logic

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 $x = 0,\ y = 6$

Types of Logic

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

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

Epistemological commitment: what states of knowledge?

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

Model

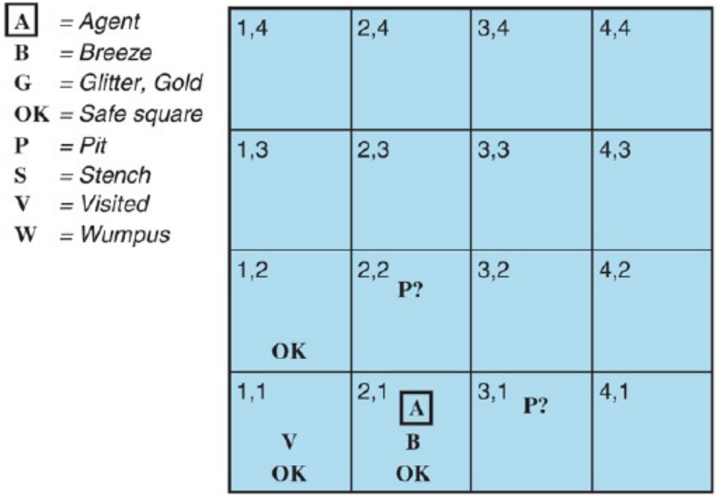
- We use the term model in place of “possible world”
- Models are mathematical abstractions, each of which has a fixed truth value (true or false) for every relevant sentence
- If a sentence α is true in model m, we says that m satisfies α or sometimes m is a model of α
- We use the notation $M(\alpha)$ to mean the set of all models of α

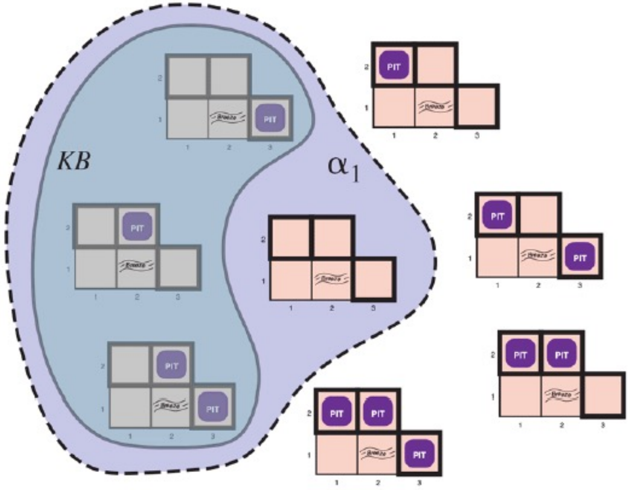
Entailment

- $\alpha \vDash \beta$ if and only if, in every model in which α is true, β is also true
    - α is a stronger assertion than β: it rules out more possible worlds
- $\alpha \vDash \beta$ if and only if $M(\alpha) \subseteq M(\beta)$
    - We are happy with the idea that the sentence $x = 0$ entails the sentence $xy = 0$
    - Obviously, in any model where x is zero, it is the case that xy is zero, regardless of the value of y
- Let's apply the same kind of logic to Wumpus-world

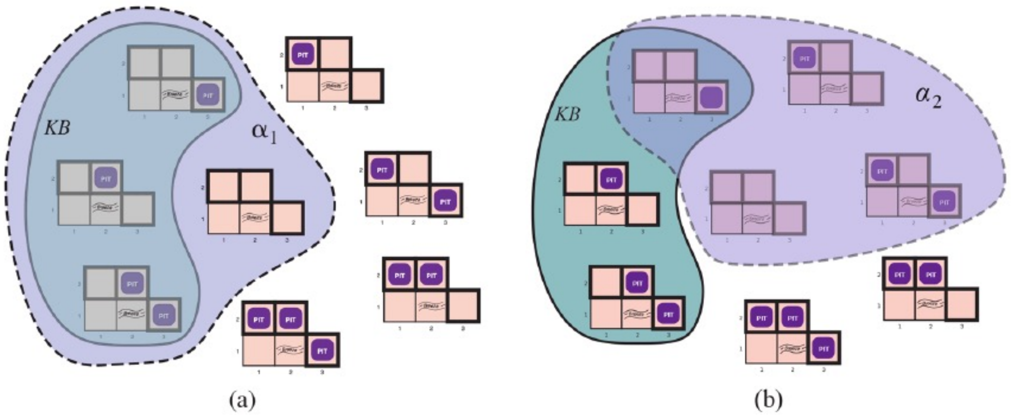
Entailment

- Nothing in [1,1]
- Breeze in [2,1]
- These percepts constitutes KB
- Do [1,2], [2,2], and [3,1] contain pits?
    - Each square might or might not contain a pit
    - $2^3 = 8$ possible models
- KB = current knowledge base
- α_1 = "There is no pit in [1,2]"





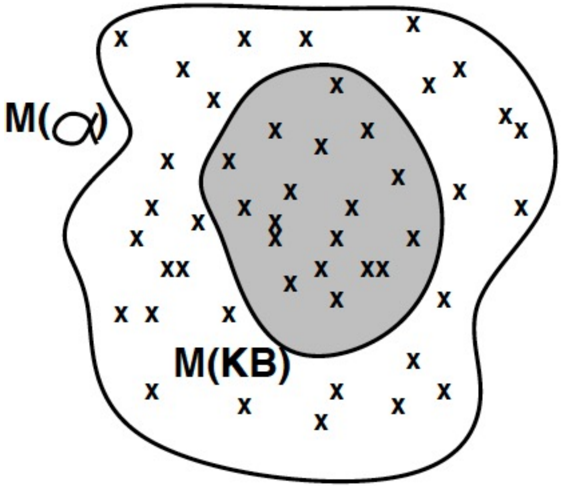
Entailment



Possible models for the presence of pits in squares [1,2], [2,2], and [3,1]. The KB corresponding to the observations of nothing in [1,1] and a breeze in [2,1] is shown by the solid line. (a) Dotted line shows models of α_1 (no pit in [1,2]). (b) Dotted line shows models of α_2 (no pit in [2,2]).

- α_1 = "There is no pit in [1,2]"
- α_2 = "There is no pit in [2,2]"
- KB = current knowledge base
- Models of α_1 are shown in (a)
- Models of α_2 are shown in (b)
- Facts:
  - every model in which KB is true, α_1 is also true: $KB \vDash \alpha_1$, i.e., there is no pit in [1,2]
  - In some models in which KB is true, α_2 is false: the agent cannot conclude that there is no pit in [2,2]; it cannot even conclude that there is a pit in [2,2]
- This example illustrates entailment, and also how to carry out logical inference
- The inference algorithm illustrated in the figure is called model checking, because it enumerates all possible models to check that α is true in all models in which KB is true: $M(KB) \subseteq M(\alpha\}$

Entailment



$$KB \models \alpha$$

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 Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"

Logical Inference

- Think of the set of all consequences of KB as the haystack and α as a needle
- Entailment is like the needle being in the haystack; inference is like finding it

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

<u>Soundness</u>: i is sound if
whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

<u>Completeness</u>: i is complete if
whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

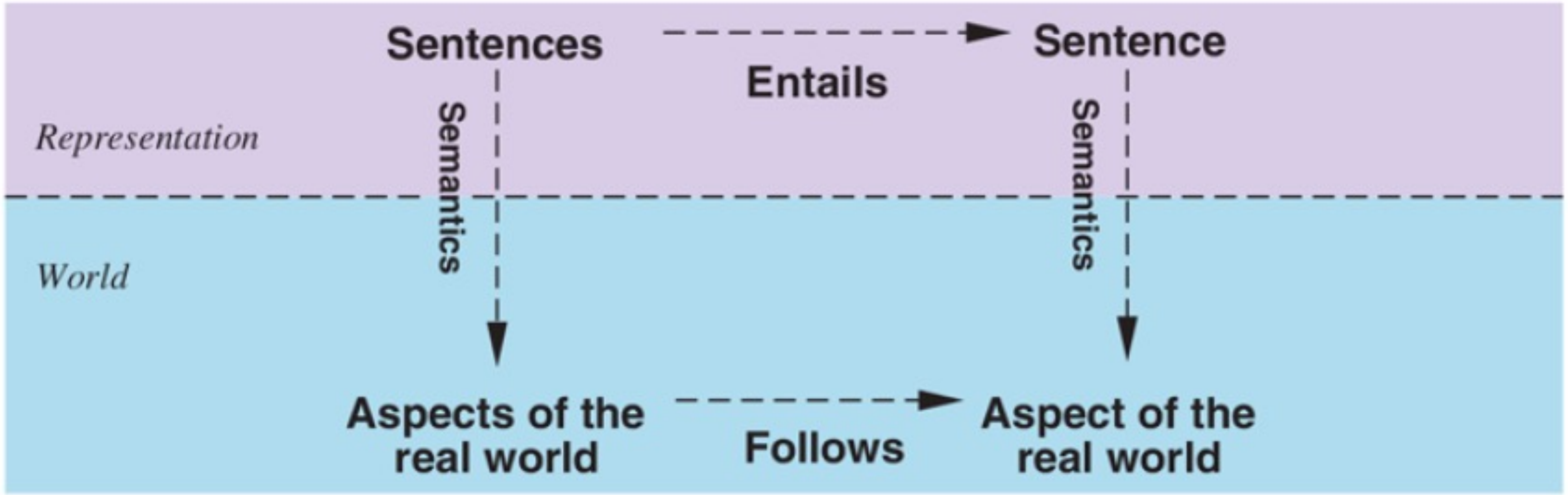
Logical Inference

- Sound or truth-preserving:
  - An unsound inference procedure announces the discovery of nonexistent needles
  - Whenever $KB \vdash_i \alpha$ it is also true that $KB \vDash \alpha$, i.e., $M(KB) \subseteq M(\alpha)$
  - If KB is true in the real world, then any sentence α derived from KB by a sound inference procedure is also true in the real world
- Completeness:
  - The inference algorithm can derive any sentence that is entailed
  - Whenever $KB \vDash \alpha$, it is also true that $KB \vdash_i \alpha$
- For finite haystack, the algorithm can decide whether the needle is in the haystack
- For many knowledge bases, however, we have infinite haystack. Completeness becomes an important issue
- Fortunately, there are complete inference procedures for logics that are sufficiently expression handle many knowledge bases.

Logical Inference

- We have described a reasoning process whose conclusions are guaranteed to be true in any world in which the premises are true
    - In particular, if KB is true in the real world, then any sentence derived from KB by a sound inference procedure is also true in the real world
- So, while an inference process operates on “syntax” – internal physical configurations such as bits in registers or patterns of electrical blips in brains – the process corresponds to the real-world relationship
- This correspondence between world and representation

Logical Inference



Sentences are physical configurations of the agent, and reasoning is a process of constructing new physical configurations from old ones. Logical reasoning should ensure that the new configurations represent aspects of the world that actually follow from the aspects that the old configurations represent.

Propositional Logic: Syntax

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols P_1, P_2 etc are sentences

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

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

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

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

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

Propositional Logic: Semantics

Each model specifies true/false for each proposition symbol

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

Rules for evaluating truth with respect to a model m:

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

Propositional Inference: Enumeration Method

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

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

Check all possible models—α must be true wherever KB is true

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

Propositional Inference: Enumeration Method

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

Validity and Satisfiability

A sentence is valid if it is true in all models

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

Validity is connected to inference via the Deduction Theorem:

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

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:

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

i.e., prove α by *reductio ad absurdum*

the form of argument that attempts to establish a claim by showing that the opposite scenario would lead to absurdity or contradiction

Proof Methods

Proof methods divide into (roughly) two kinds:

<u>Model checking</u>

- truth table enumeration (sound and complete for propositional)
- heuristic search in model space (sound but incomplete)
  - e.g., the GSAT algorithm (Ex. 6.15)

<u>Application of inference rules</u>

- 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 alg.

Inference Rules for Propositional Logic

Modus Ponens (Implication-Elimination)

From an implication and its premise, infer conclusion:

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

And-Elimination

From a conjunction, you can infer any conjunct

$$\frac{\alpha_1 \land \alpha_2 \land \alpha_3 \land \ldots \land \alpha_n}{\alpha_i}$$

Inference Rules for Propositional Logic

And-Introduction

From a list of sentences, you can infer the conjunct

$$\frac{a_1, a_2, a_3, \ldots, a_n}{a_1 \land a_2 \land a_3 \land \ldots \land a_n}$$

Or-Introduction

From a sentence, infer its disjunction with anything

$$\frac{a_i}{a_1 \lor a_2 \lor a_3 \lor \ldots \lor a_n}$$

Inference Rules for Propositional Logic

Double-Negative Elimination

From a double negation, infer the positive sentence

$$\frac{\neg\neg a}{a}$$

Unit Resolution

From a disjunction in which one is false, then you can infer the other is true

$$\frac{a \vee \beta, \neg\beta}{a}$$

Inference Rules for Propositional Logic

Resolution

Since beta cannot be both true and false, one of the disjuncts must be true

$$\frac{a \vee \beta, \neg\beta \vee \gamma}{a \vee \gamma}$$

Implication is transitive

$$\frac{\neg a \Rightarrow \beta, \beta \Rightarrow \gamma}{\neg a \Rightarrow \gamma}$$

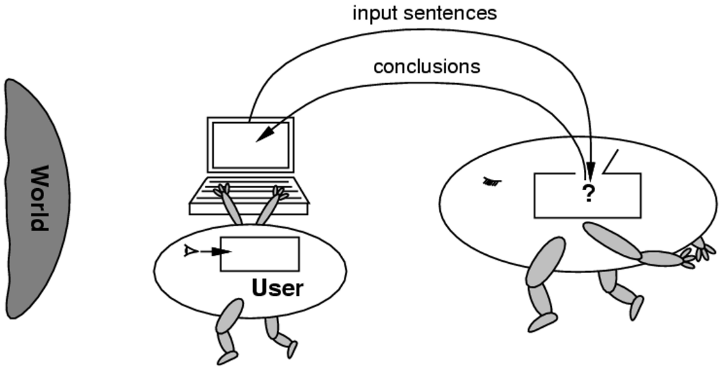
Truth Table for Resolution

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*
<u>*False*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>
<u>*True*</u>	<u>*False*</u>	<u>*False*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>
<u>*True*</u>	<u>*False*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>
True	*True*	*False*	*True*	*False*	*True*
<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>	<u>*True*</u>

Truth tables can also be used to verify the inference rules

$$\frac{a \vee \beta\, ,\, \neg\beta \vee \gamma}{a \vee \gamma}$$

Logical Agents



Input sentences can come from the user perceiving the world, or from a machine-readable representation of the world

Infer new statements about the world that are valid

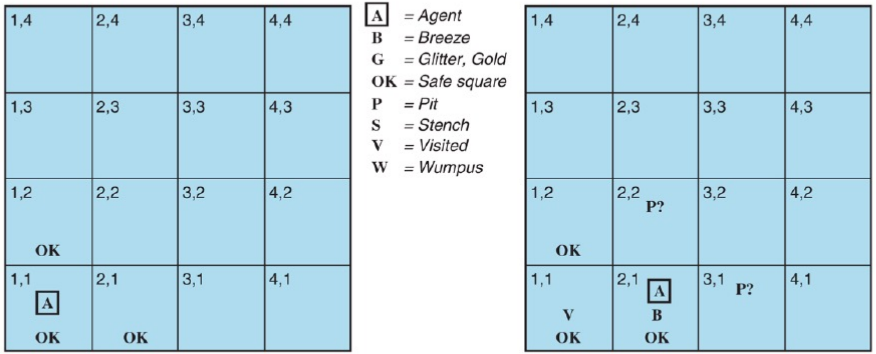
Exercise: An Agent for the Wumpus World

Convert perceptions into sentences:
"In square [1,1], there is no breeze and no stench" ... becomes...

$$\neg B_{11} \land \neg S_{11}$$

Start with some knowledge of the world (in the form of rules)
R1 : $\neg S_{11} \Rightarrow \neg W_{11} \land \neg W_{12} \land \neg W_{21}$
R2 : $\neg S_{21} \Rightarrow \neg W_{11} \land \neg W_{21} \land \neg W_{22} \land \neg W_{31}$
....
R4 : $S_{12} \Rightarrow W_{13} \lor W_{12} \lor W_{22} \lor W_{11}$

What can we derive from Percepts: $\neg S_{11}$, $\neg S_{21}$, S_{12}?



Finding the Wumpus

Percepts: $\neg S_{11}$ $\neg S_{21}$ S_{12}

1. Apply modus ponens and and-elimination to $\neg S_{11}$ to get:
   $\neg W_{11}$ $\neg W_{12}$ $\neg W_{21}$
2. Apply modus ponens and and-elimination to $\neg S_{21}$ to get:
   $\neg W_{22}$ $\neg W_{21}$ $\neg W_{31}$
3. Apply modus ponens to S_{12} and R_4:
   $W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$
4. Apply unit resolution to #3 and #1
   $W_{13} \vee W_{22}$
5. Apply unit resolution to #4 and #2
   W_{13}

The wumpus is in square [1,3]!!!

Problems with Propositional Logic

Too many propositions!

- How can you encode a rule such as "don't go forward if the wumpus is in front of you"?
- In propositional logic, this takes (16 squares * 4 orientations) = 64 rules!

Truth tables become unwieldy quickly

- Size of the truth table is 2^n where *n* is the number of propositional symbols

More Problems with Propositional Logic

No good way to represent changes in the world

- How do you encode the location of the agent?

What kinds of practical applications is this good for?

- Relatively little

Nevertheless, In Summary

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

Basic concepts of logic:

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

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

Propositional logic suffices for some of these tasks

Truth table method is sound and complete for propositional logic