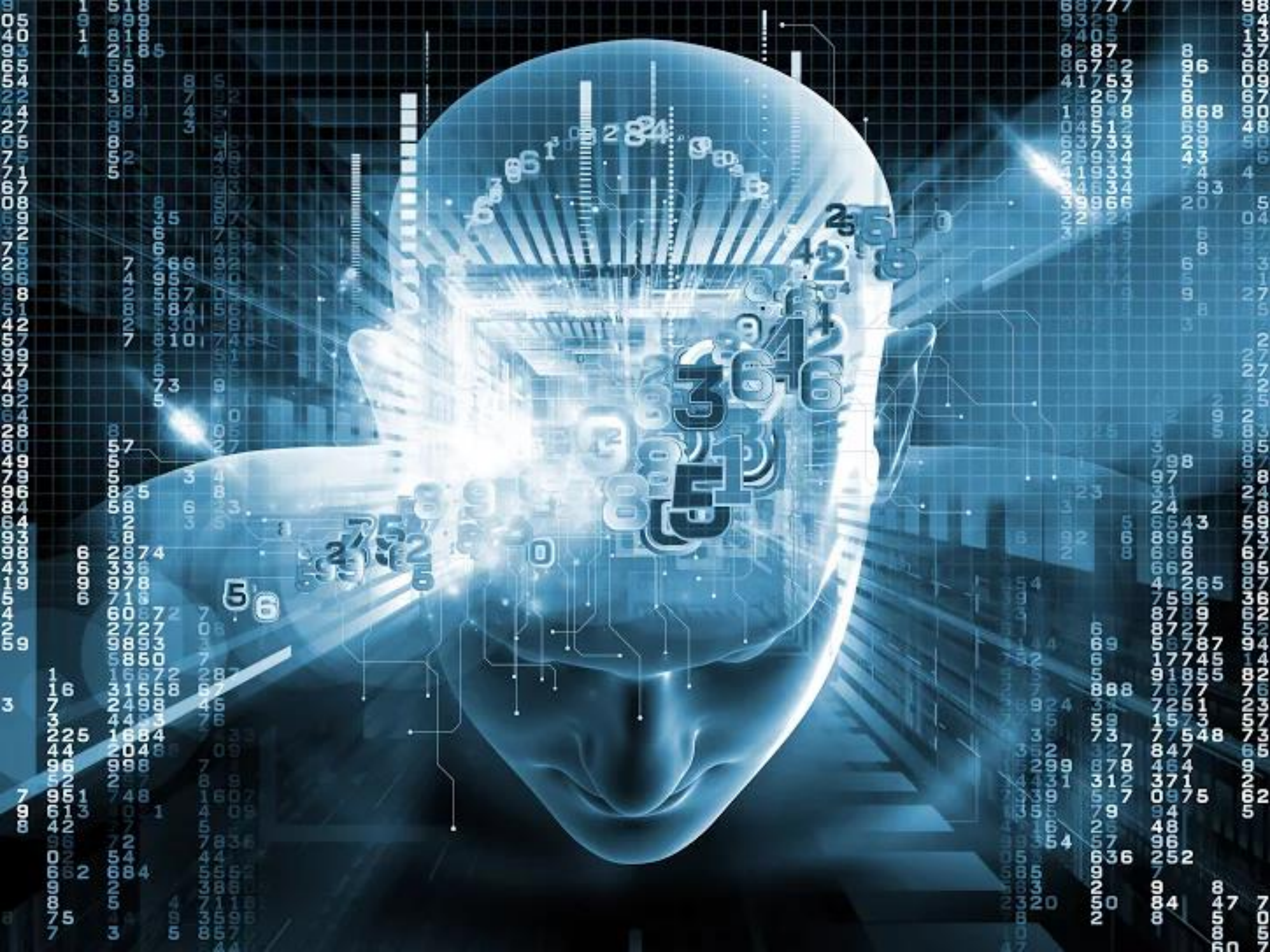


Knowledge-based agents

- 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**.

Knowledge-based agents

- 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 executes the chosen action.

Architecture of Knowledge

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

- The level at which the knowledge is encoded into sentences.
- Example: Links(GoldenGateBridge, SanFrancisco, MarinCounty).

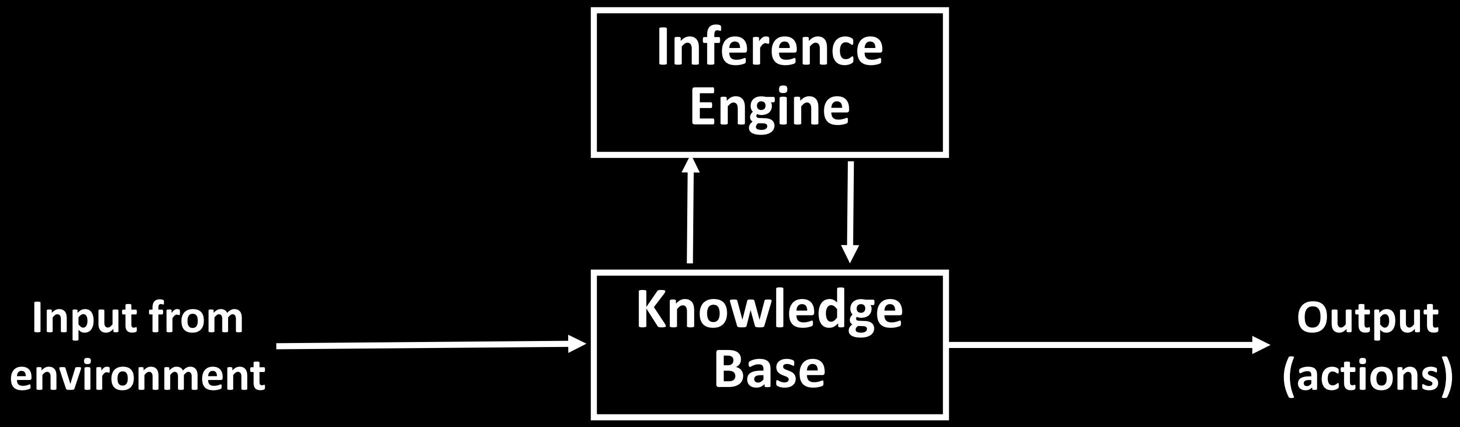
Implementation Level

- The physical representation of the sentences in the logical level.
- Example: ‘`(links goldengatebridge sanfrancisco marincounty)`

Inference

- The **Inference Engine** derives new sentences from the input and KB
- The inference mechanism depends on representation in KB
- The agent operates as follows:
  1. It receives percepts from environment
  2. It computes what action it should perform (by IE and KB)
  3. It performs the chosen action (some actions are simply inserting inferred new facts into KB).

Architecture of Knowledge



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.
- Occasionally, there is a heap of gold in a room.
- The goal is:
  - to collect the gold and
  - exit the world
  - Don’t get eaten

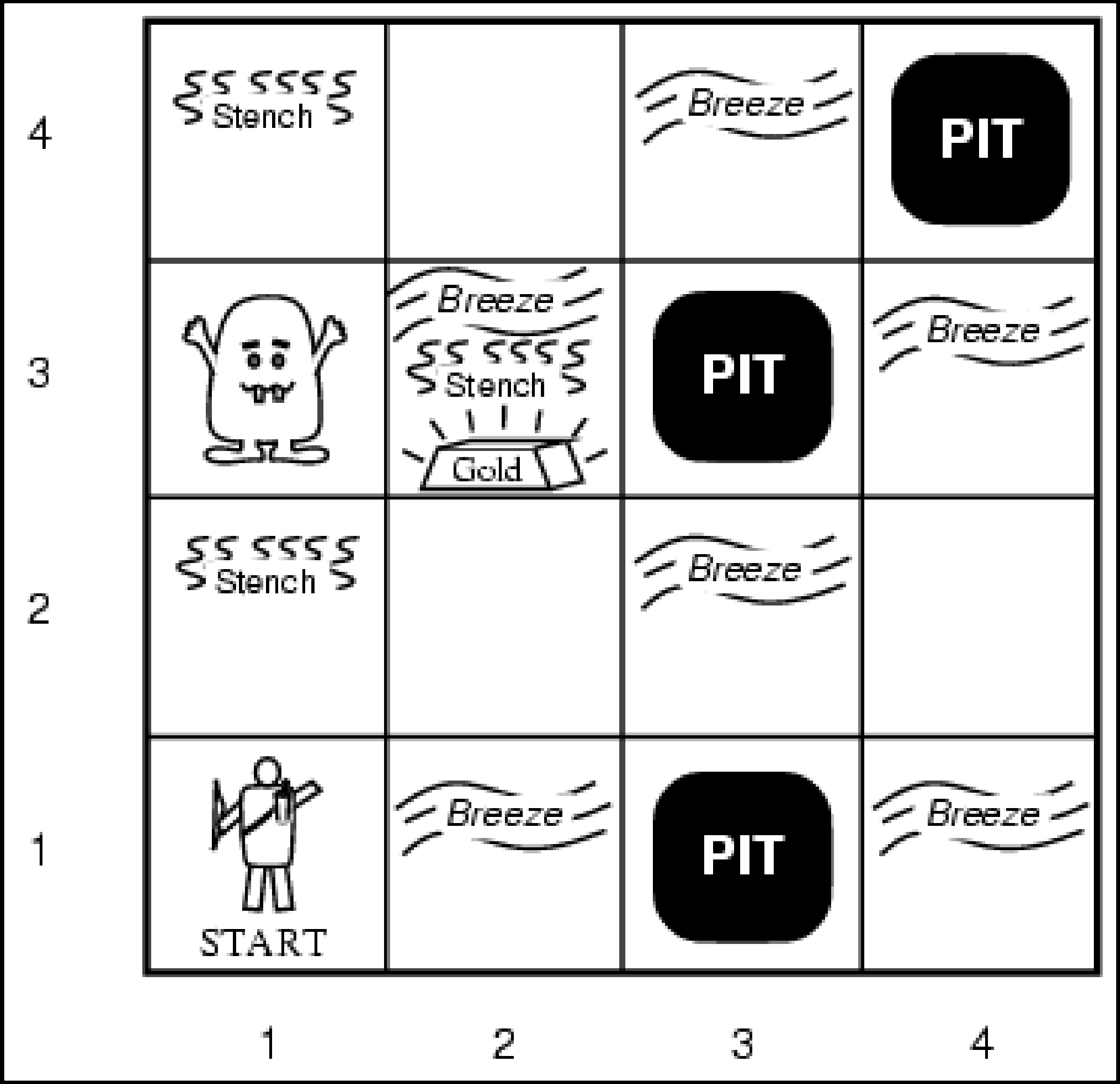
History of "Hunt the Wumpus"

- **WUMPUS** /wuhm'p*s/ n. The central monster (and, in many versions, the name) of a famous family of very early computer games called "Hunt The Wumpus," dating back at least to 1972
- The wumpus lived somewhere in a cave with the topology of a dodecahedron's edge/vertex graph
  - (later versions supported other topologies, including an icosahedron and Mobius strip).
- The player started somewhere at random in the cave with five "crooked arrows";
  - these could be shot through up to three connected rooms, and would kill the wumpus on a hit
    - (later versions introduced the wounded wumpus, which got very angry).

A typical Wumpus world

The agent starts in the field [1,1].

The 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 will be 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 actions of the agent in Wumpus game are:

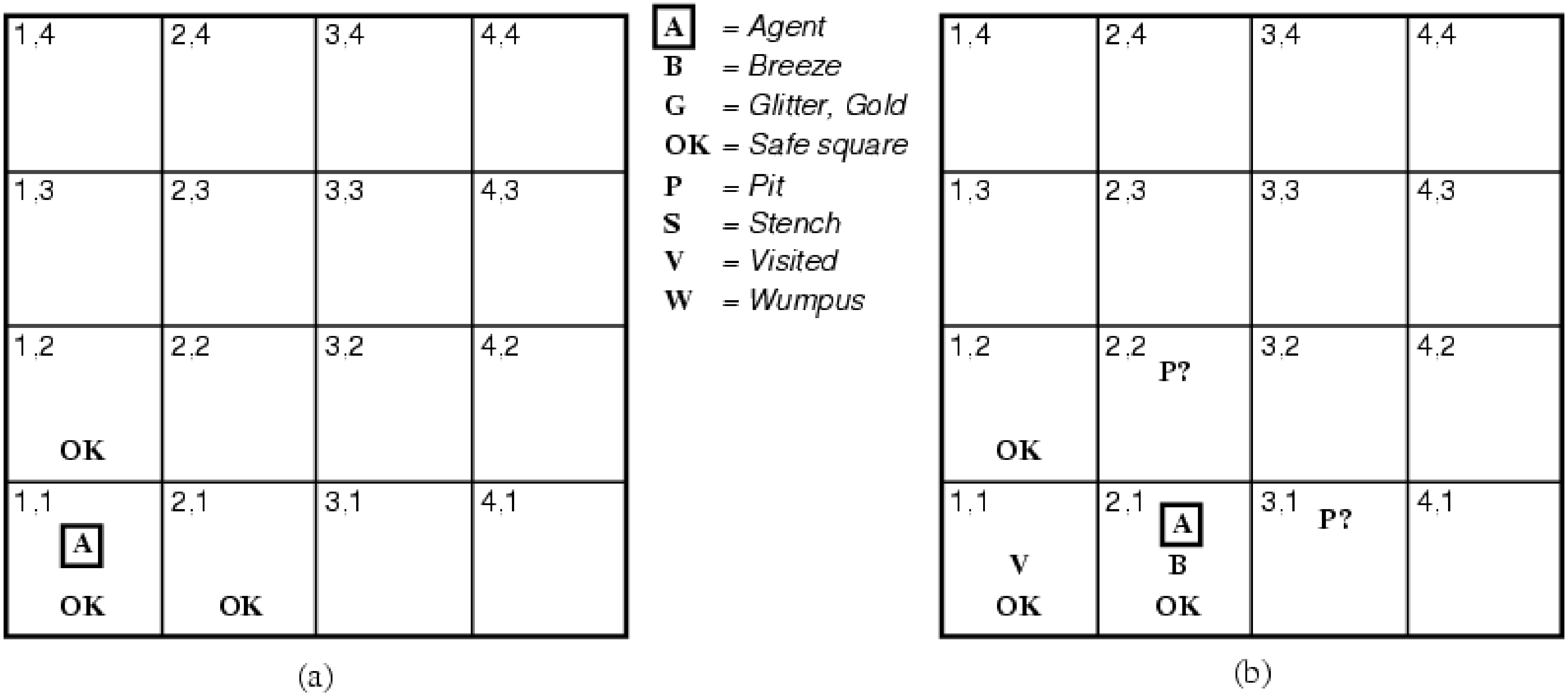
- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab** means pick up an object that is in the same square as the agent
- **shoot** means fire an arrow in a straight line in the direction the agent is looking.
  - The arrow continues until it either hits and kills the wumpus or hits the wall.
  - The agent has only one arrow.
  - Only the first shot has any effect.
- **climb** is used to leave the cave.
  - Only effective in start field.
- **die**, if the agent enters a square with a pit or a live wumpus.
  - (No take-backs!)

The agent's goal

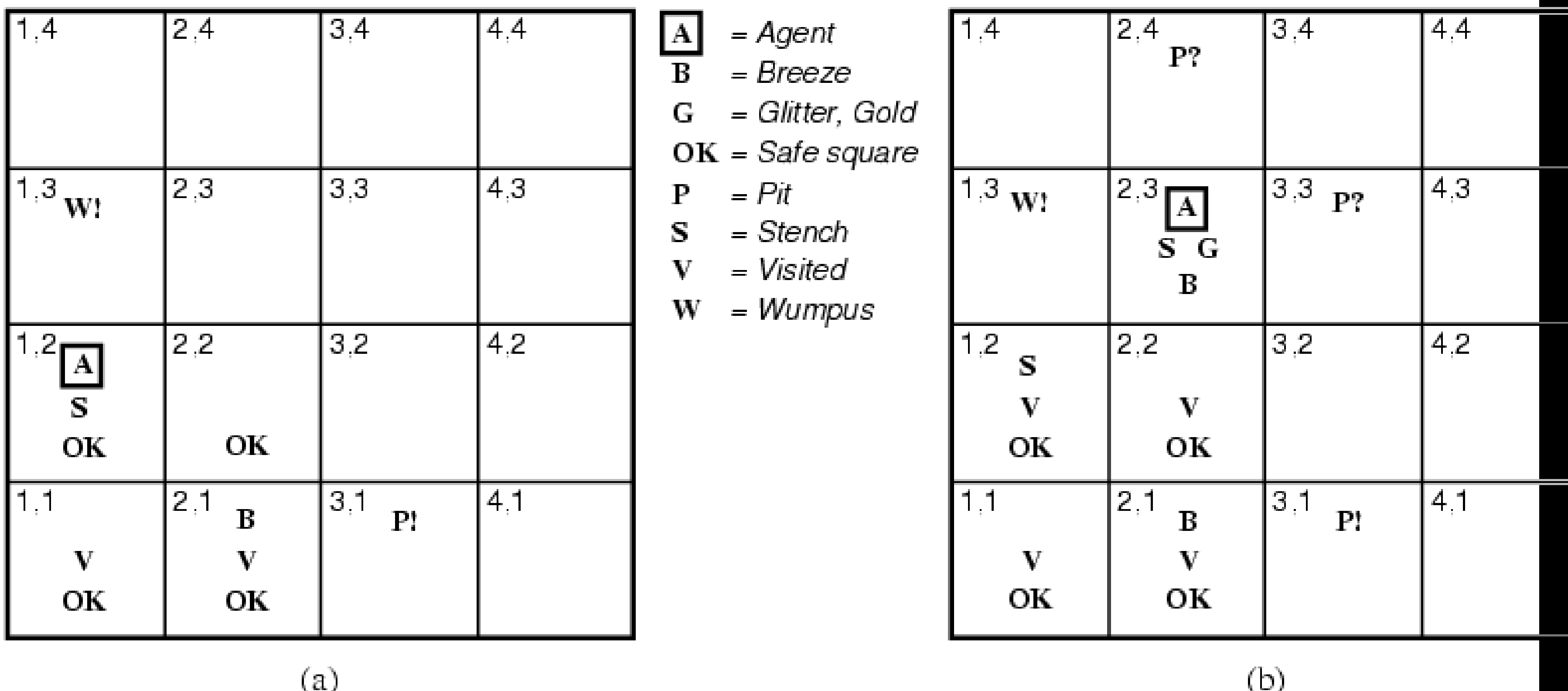
The agent's goal is to find the gold and bring it back to the start as quickly as possible, without getting killed.

- 1000 points reward for climbing out of the cave with the gold
- 1 point deducted for every action taken
- 10000 points penalty for getting killed
- 100 points for killing the Wumpus

The Wumpus agent's first step



Later

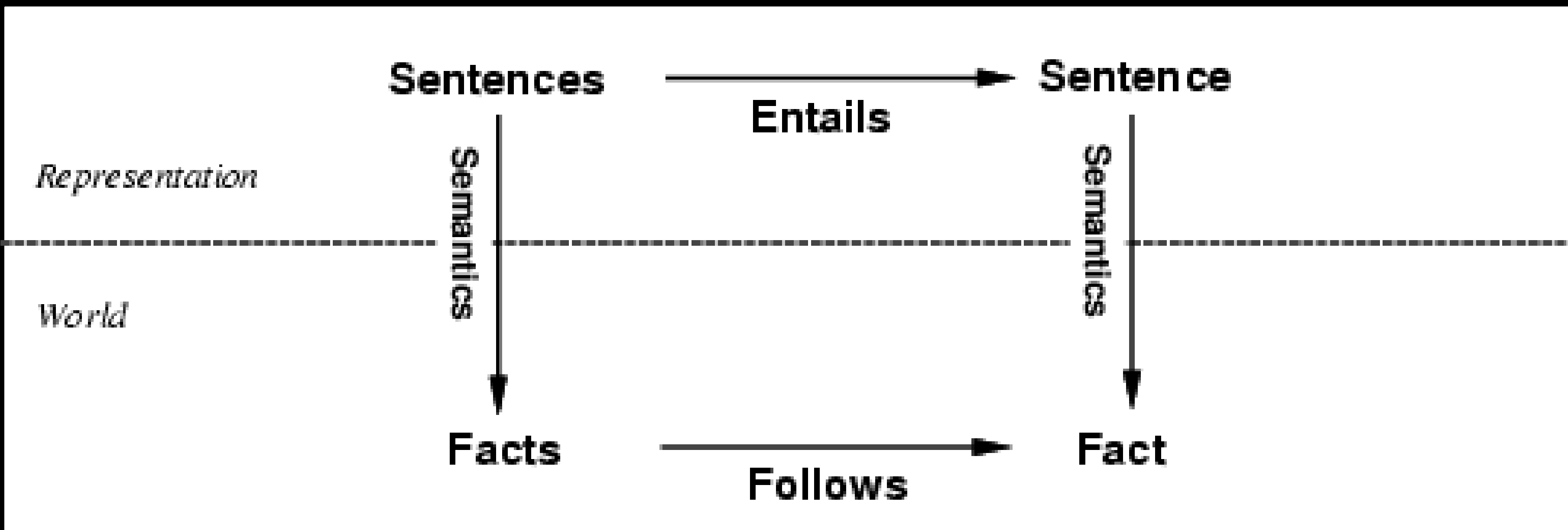


1,4
2,4
3,4
4,4
1,3 W!
2,3
3,3
4,3
1,2 A S OK
2,2 OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P!
4,1
(a)
A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus
1,4
2,4 P?
3,4
4,4
1,3 W!
2,3 A S G B
3,3 P?
4,3
1,2 S V OK
2,2 V OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P!
4,1
(b)

Representation, reasoning, and logic

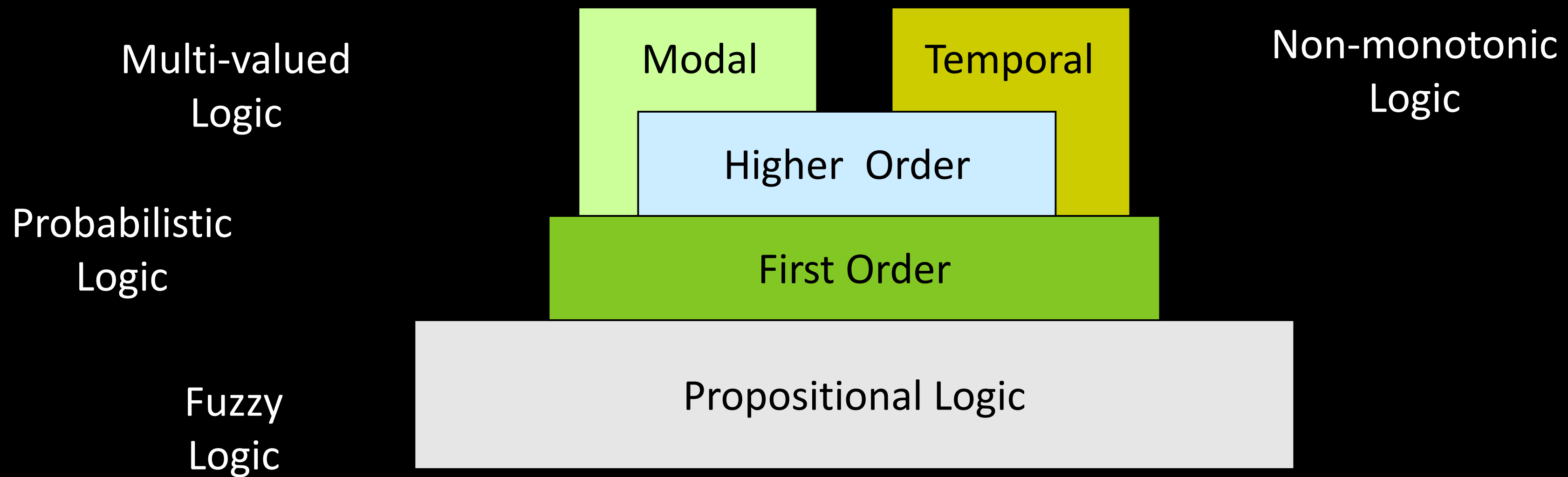
- The object of *knowledge representation* is to express knowledge in a **computer-tractable form**, so that agents can perform well.
- A knowledge representation language is defined by:
  - its **syntax**, which defines all possible sequences of symbols that constitute sentences of the language.
    - Examples: 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 <u>maps sentences</u> in logic <u>to facts</u> in the world.
The property of *one fact following from another* is mirrored by the property of <u>*one sentence being entailed by another*</u>.

Different Logics



Different Logics

Propositional Logic: Sentences are atomic in form.

A,B,C A->B, AorB

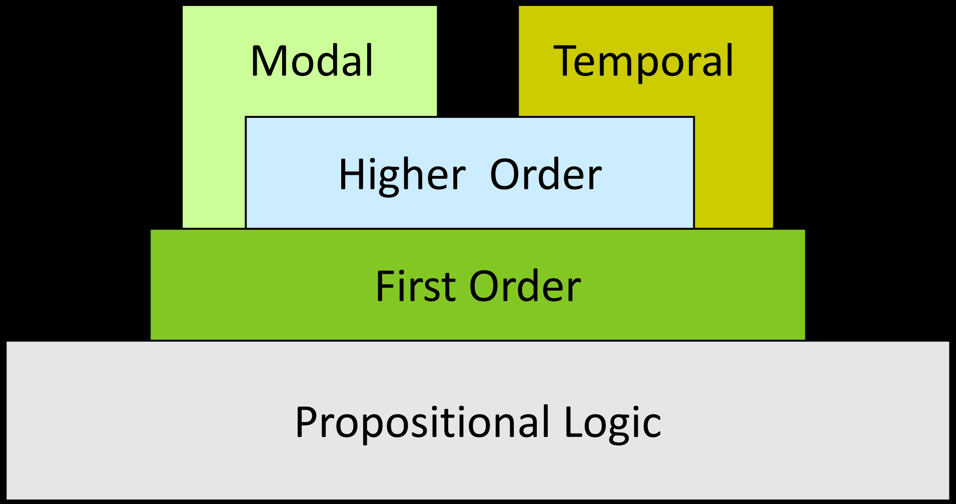
First Order PC: Sentences are predicates applied to objects

(on A B), (taller Bob Sam), if(on ?x ?y) (covered ?y)

Higher Order: Quantification can be applied to features

Modal Logic: Sentences can be qualified by features like certainty

Temporal Logic: Sentences are linked to time.



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 branch of philosophy that concerns the *forms, nature, and preconditions of knowledge.*

Problems with ontologies

They get complex

They aren’t consistent

They change over time

They aren’t everything *

Propositional logic

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

  $\wedge$ ...and

  $\vee$ ...or

  $\Rightarrow$...implies

  $\Leftrightarrow$..is equivalent

  $\neg$ ...not

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 of these symbols, e.g.:
  - P means "It is hot"
  - Q means "It is humid"
  - R means "It is raining"
- A sentence (aka formula, well-formed formula, wff) defined as:
  - A symbol
  - If S is a sentence, then ~S is a sentence (e.g., "not")
  - If S is a sentence, then so is (S)
  - If S and T are sentences, then (S v T), (S ^ T), (S => T) , and (S <=> T) are sentences (e.g., "or," "and," "implies," and "if and only if")
  - A finite number of applications of the above

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"

A BNF grammar of sentences in propositional logic

```
S := <Sentence> ;
<Sentence> := <AtomicSentence> | <ComplexSentence> ;
<AtomicSentence> := "TRUE" | "FALSE" |
                    "P" | "Q" | "S" ;
<ComplexSentence> := "(" <Sentence> ")" |
          <Sentence> <Connective> <Sentence> |
          "NOT" <Sentence> ;
<Connective> := "NOT" | "AND" | "OR" | "IMPLIES" |
  "EQUIVALENT" ;
```

The overall model

- The meaning or **semantics** of a sentence determines its **interpretation**.
- Given the truth values of all of 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" in which each sentence in the KB is True.
- 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

And

p	q	$p \cdot q$
T	T	T
T	F	F
F	T	F
F	F	F

Or

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

If . . . then

p	q	$p \supset q$
T	T	T
T	F	F
F	T	T
F	F	T

Not

p	$\sim p$
T	F
F	T

Truth tables II

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*

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*

Inference rules

- **Logical inference** is used to create new sentences that logically follow from a given set of sentences (KB).
- An inference rule is **sound** if every sentence X produced by it 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 the KB.
  - We also say - *"expression is entailed by KB"*.
  - Pleas note the analogy to complete search algorithms.

Sound rules of inference

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

RULE	PREMISE	CONCLUSION
Modus Ponens	A, A => B	B
And Introduction	A, B	A ^ B
And Elimination	A ^ B	A
Double Negation	~~A	A
Unit Resolution	A v B, ~B	A
Resolution	A v B, ~B v C	A v C

Sound Inference Rules (deductive rules)

- Here are some examples of **sound rules of inference**.
- Each can be shown to be sound using a truth table -- a rule is sound if it’ s conclusion is true whenever the premise is true.

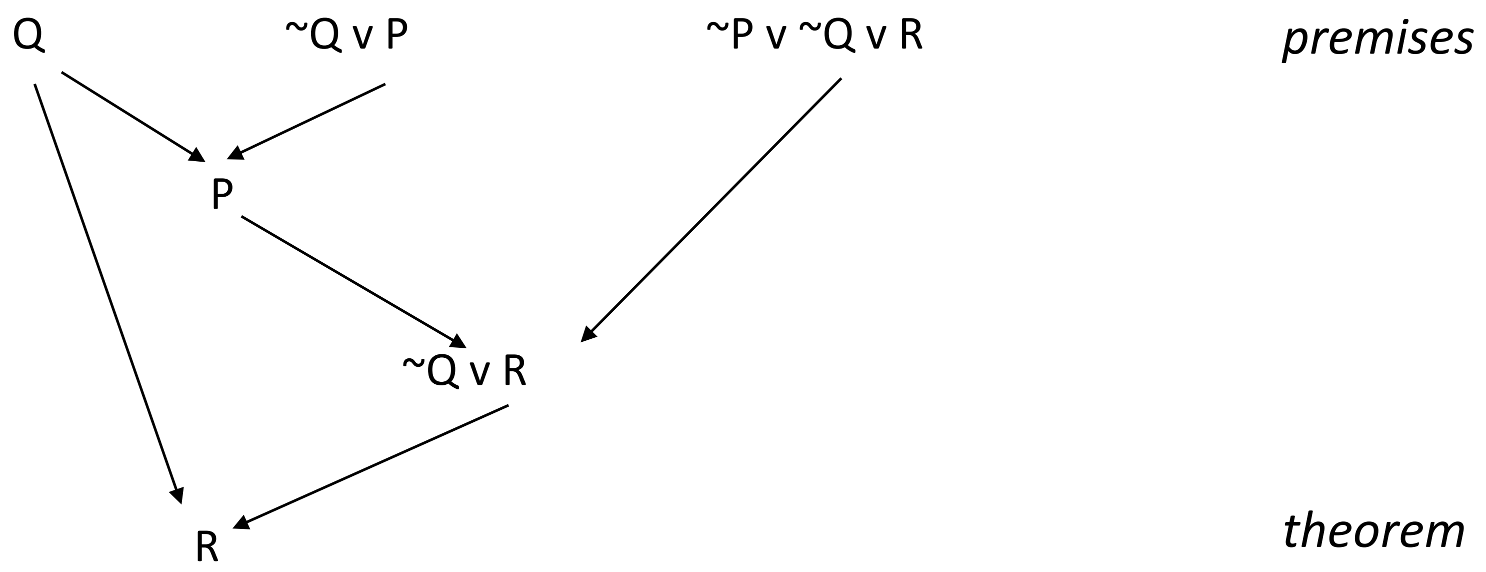
RULE	PREMISE	CONCLUSION
Modus Tollens	~B, A => B	~A
Or Introduction	A	A v B
Chaining	A => B, B => C	A => C

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.

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(1,2)	"It is hot and humid"
6	R	Modus Ponens(4,5)	"It is raining"

Proof by resolution



- Theorem proving as search
  - **Start node:** the set of given premises/axioms (KB + Input)
  - **Operator:** inference rule (add a new sentence into parent node)
  - **Goal:** a state that contains the theorem asked to prove
  - **Solution:** a path from start node to a goal

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

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,
  - or **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,
  - or **all valid sentences can be proved from the premises.**

Propositional logic is Weak

- Hard to identify "**individuals**." E.g., Mary, 3
  - –Individuals cannot be PL sentences themselves.
- Can't directly talk about properties of individuals or relations between individuals. (hard to connect individuals to class properties).
  - –E.g., property of being a human implies property of being mortal
  - –E.g. "Bill is tall"
- Generalizations, patterns, regularities can't easily be represented.
  - –E.g., all triangles have 3 sides
  - –All members of a class have this property
  - –Some members of a class have this property
- A better representation is needed to capture the relationship (and distinction) *between objects and classes*, including properties belonging to classes and individuals.

Confusius Example: weakness of PL

- 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?

What do we need

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

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

- so the above 3 sentences are represented as:

  P => Q;   R => P;   R => 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 means for representing the fact that all individuals who are "people" are also "mortal."

What do we need

- First-Order Logic (abbreviated FOL or FOPC) is expressive enough to concisely represent this kind of situation by separating classes and individuals
  - Explicit representation of individuals and classes, x, Mary, 3, persons.
  - Adds relations, variables, and quantifiers, e.g.,
    - *"Every person is mortal"* Forall X: person(X) => mortal(X)
    - *"There is a white alligator"* There exists some X: Alligator(X) ^ white(X)

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)
  W22 = The Wumpus is in cell (2,2)
  V11 = We have visited cell (1,1)
  OK11 = Cell (1,1) is safe.
  etc

- Some rules

  (R1) ~S11 => ~W11 ^ ~W12 ^ ~W21
  (R2) ~S21 => ~W11 ^ ~W21 ^ ~W22 ^ ~W31
  (R3) ~S12 => ~W11 ^ ~W12 ^ ~W22 ^ ~W13
  (R4) S12 => W13 v W12 v W22 v W11
  etc

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

Problems with the propositional Wumpus hunter

- Lack of variables prevents stating more general rules.
  - E.g., 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.

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 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 this fact is part of the actual world, then the sentence is true.

- The process of deriving new sentences from old one 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 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 write 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 a simple semantic. It suffices to illustrate the process of inference.
  - Propositional logic quickly becomes impractical, even for very small worlds.

Last Time: Propositional Logic

alarm ^ nighttime => burglar
stars => nighttime
nighttime => dark
dark => nighttime
burglar => crime
crime ^ dark => unsafe
alarm => noise
noise ^ nighttime => annoyed-neighbors
alarm
dark

Prove that this neighborhood is unsafe the above KB of facts

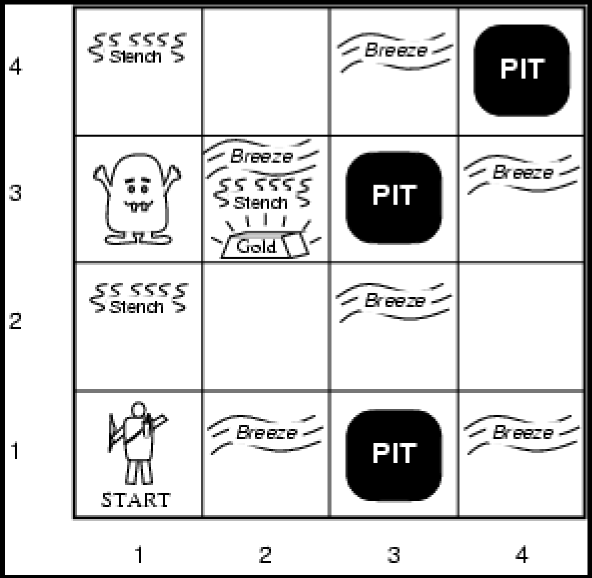
Problems with Propositional Logic

Impossible to make general assertions

"Pits cause breezes in adjacent squares"

$B2,1 \Leftrightarrow (P1,1 \vee P2,2 \vee P3,1)$

$P3,1 \Leftrightarrow (B2,1 \wedge B3,2 \wedge B4,1)$



Pros and cons of propositional logic

- ☺ Propositional logic is declarative
- ☺ Propositional logic allows partial/disjunctive/negated information
  - (unlike most data structures and databases)
- ☺ Propositional logic is compositional:
  - meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- ☺ Meaning in propositional logic is context-independent
  - (unlike natural language, where meaning depends on context)

- ☹ Propositional logic has very limited expressive power
  - (unlike natural language)
  - E.g., cannot say "pits cause breezes in adjacent squares“
    - except by writing one sentence for each square
    -

- Propositional Logic
  - Is simple
  - Illustrates important points:
    - Model, satisfiability, inference
  - Is restrictive: world is a set of facts
  - Lacks expressiveness (world contains FACTS)
- First-Order Logic
  - More symbols (objects, properties, relations)
  - More connectives (quantifiers)

First-order logic

- Whereas propositional logic assumes the world contains facts,
- first-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, colors, baseball games, wars, …
  - Relations: red, round, prime, brother of, bigger than, part of, comes between, …
  - Functions: father of, best friend, one more than, plus, …
  -

Propositional Logic vs. FOL/FOPC

- Propositional Logic
  - The world consists of propositions (sentences) which can be true or false.
- Predicate Calculus (First Order Logic)
  - The world consists of objects, functions and relations between the objects.

Syntax of FOL: Basic elements

- Constants KingJohn, 2, NUS,...
- Predicates Brother, >,...
- Functions Sqrt, LeftLegOf,...
- Variables x, y, a, b,...
- Connectives $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
- Equality =
- Quantifiers $\forall, \exists$

Atomic sentences

Atomic sentence = *predicate* (*term*$_1$,...,*term*$_n$) or *term*$_1$ = *term*$_2$

Term = *function* (*term*$_1$,...,*term*$_n$) or *constant* or *variable*

- E.g., *Brother(KingJohn,RichardTheLionheart) > (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))*

Complex sentences

- Complex sentences are made from atomic sentences using connectives
-

$$\neg S,\ S_1 \wedge S_2,\ S_1 \vee S_2,\ S_1 \Rightarrow S_2,\ S_1 \Leftrightarrow S_2,$$

E.g. $Sibling(KingJohn,Richard) \Rightarrow Sibling(Richard,KingJohn)$

$>(1,2) \vee \leq (1,2)$

$>(1,2) \wedge \neg >(1,2)$

Universal quantification

- $\forall$*<variables> <sentence>*
-

Everyone at NU is smart:
$\forall$x At(x,NU) $\Rightarrow$ Smart(x)

- $\forall$x *P* is true in a model *m* iff *P* is true with *x* being each possible object in the model
-
- Roughly speaking, equivalent to the conjunction of instantiations of *P*

  At(KingJohn,NU) $\Rightarrow$ Smart(KingJohn)
  $\wedge$ At(Richard,NU) $\Rightarrow$ Smart(Richard)
  $\wedge$ At(Jane,NU) $\Rightarrow$ Smart(Bob)
  $\wedge$ ...

A common mistake to avoid

- Typically, $\Rightarrow$ is the main connective with $\forall$
-
- Common mistake: using $\wedge$ as the main connective with $\forall$:

  $\forall x\ At(x,NU) \wedge Smart(x)$

  means "Everyone is at NU and everyone is smart"

Existential quantification

- $\exists$<*variables*> <*sentence*>
- Someone at NU is smart:
- $\exists x$ At(x,NU) $\wedge$ Smart(x)
-
- $\exists x$ *P* is true in a model *m* iff *P* is true with *x* being some possible object in the model
-
- Roughly speaking, equivalent to the disjunction of instantiations of *P*
-

  At(KingJohn,NU) $\wedge$ Smart(KingJohn)
  $\vee$ At(Richard,NU) $\wedge$ Smart(Richard)
  $\vee$ At(Jane,NU) $\wedge$ Smart(NU)
  $\vee$ ...

Another common mistake to avoid

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:
-

$$\exists x \text{ At}(x,NU) \Rightarrow \text{Smart}(x)$$

is true if there is anyone who is not at NU!

Properties of quantifiers

- $\forall x \, \forall y$ is the same as $\forall y \, \forall x$
- $\exists x \, \exists y$ is the same as $\exists y \, \exists x$
- $\exists x \, \forall y$ is not the same as $\forall y \, \exists x$
- $\exists x \, \forall y \, Loves(x,y)$
  - "There is a person who loves everyone in the world"
- $\forall y \, \exists x \, Loves(x,y)$
  - "Everyone in the world is loved by at least one person"
- Quantifier duality: each can be expressed using the other
- $\forall x \, Likes(x,IceCream)$ $\qquad \neg\exists x \, \neg Likes(x,IceCream)$
- $\exists x \, Likes(x,Broccoli)$ $\qquad \neg\forall x \, \neg Likes(x,Broccoli)$

Quantifiers

- Existential:
  - There is a Northwestern Student from Hawaii.
- Universal:
  - Northwestern students live in Evanston.

Examples

- All purple mushrooms are poisonous
- No purple mushroom is poisonous
- Every CS student knows a programming language.
- A programming language is known by every CS student

Equality

- $term_1 = term_2$ is true under a given interpretation if and only if $term_1$ and $term_2$ refer to the same object
-
- E.g., definition of *Sibling* in terms of *Parent*:
-

$$\forall x,y\ Sibling(x,y) \Leftrightarrow [\neg(x = y) \land \exists m,f\ \neg (m = f) \land Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)]$$

Using FOL

The kinship domain:

- Brothers are siblings

  $\forall$x,y *Brother(x,y)* $\Leftrightarrow$ *Sibling(x,y)*

- One's mother is one's female parent

  $\forall$m,c *Mother(c)* = m $\Leftrightarrow$ *(Female(m)* $\wedge$ *Parent(m,c))*

- "Sibling" is symmetric

  $\forall$x,y *Sibling(x,y)* $\Leftrightarrow$ *Sibling(y,x)*

Interacting with FOL KBs

- Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at *t=5*:

  `Tell`(KB,Percept([Smell,Breeze,None],5))
  `Ask`(KB,∃a BestAction(a,5))

- I.e., does the KB entail some best action at *t=5*?
-
- Answer: *Yes*, {*a/Shoot*} ← substitution (binding list)
- Given a sentence *S* and a substitution σ,
- *S*σ denotes the result of plugging σ into *S*; e.g.,
  S = Smarter(x,y)
  σ = {x/Jane,y/Sue}
  *S*σ = Smarter(Jane,Sue)

- `Ask`(KB,S) returns some/all σ such that KB ╞ σ
-

Knowledge base for the wumpus world

- Perception
  - $\forall$t,s,b Percept([s,b,Glitter],t) $\Rightarrow$ Glitter(t)
  -

- Reflex
  - $\forall$t Glitter(t) $\Rightarrow$ BestAction(Grab,t)

Deducing hidden properties

- $\forall$x,y,a,b *Adjacent*([x,y],[a,b]) $\Leftrightarrow$ [a,b] $\in$ {[x+1,y], [x-1,y],[x,y+1],[x,y-1]}

Properties of squares:

- $\forall$s,t *At*(Agent,*s*,t) $\wedge$ Breeze(t) $\Rightarrow$ Breezy(s)

Squares are breezy near a pit:

- Diagnostic rule---infer cause from effect
  $\forall$s Breezy(s) $\Rightarrow$ $\exists$r Adjacent(r,s) $\wedge$ Pit(r)
- Causal rule---infer effect from cause
  $\forall$r Pit(r) $\Rightarrow$ [$\forall$s Adjacent(r,s) $\Rightarrow$ Breezy(s)]

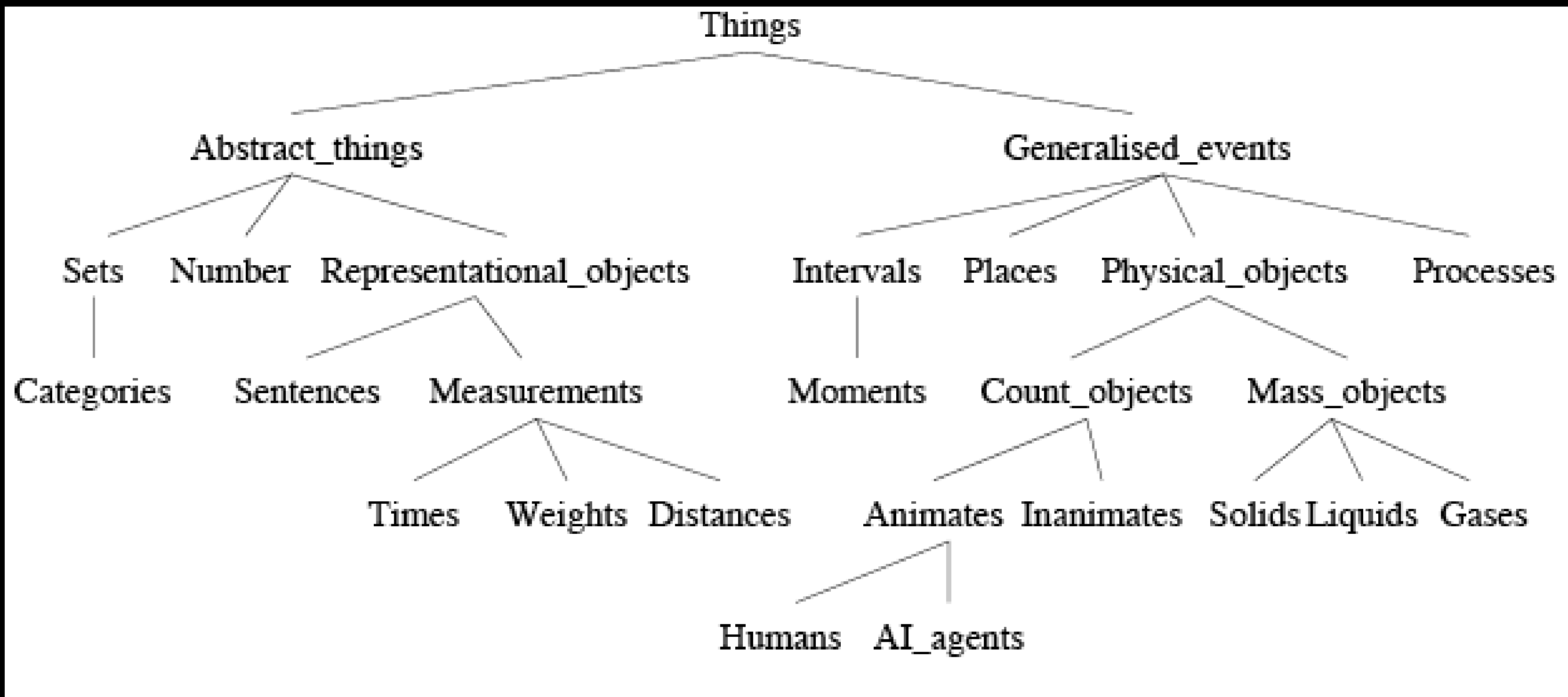
Knowledge engineering in FOL

1. Identify the task
2. Assemble the relevant knowledge
3. Decide on a vocabulary of predicates, functions, and constants
4. Encode general knowledge about the domain
5. Encode a description of the specific problem instance
6. Pose queries to the inference procedure and get answers
7. Debug the knowledge base
8.

Knowledge Representation

- Representing general concepts
  - ACTIONS
  - TIME
  - PHYSICAL OBJECTS
  - BELIEFS
- Ontological Engineering versus Knowledge Engineering

Upper Ontology



Categories and Objects

- Predicates
  - Basketball(b)
- Objects
  - Basketballs
- Inheritance
  - Every Apple is edible
- Taxonomy/Taxonomic Hierarchy

Stating facts about categories

- An object is a member of a category
- A category is a subclass of another category
- All members of a category have some properties
- Members of a category can be recognized by some properties
- A category as a whole has some properties

Categories

- Disjoint
  - Disjoint({Animals, Vegetables})
- Exhaustive Decomposition
  - ExhaustiveDecomposition({Americans, Canadians, Mexicans}, NorthAmericans)
- Partition
  - Partition({Males, Females}, Animals)

Physical Composition

- PartOf relation to relate two things
  - PartOf(Bucharest, Romania)
  - PartOf(Romanai, Eastern Europe)
  - PartOf(EasternEurope, Europe)
  - PartOf(Europe, Earth)
  - Therefore PartOf(Bucharest, Earth)
- Composite Objects
  - Biped has two legs attached to a body
  - Biped(a) => ∃ l1, l2, b Body(b) ∩ Leg(l1) ∩ Leg(l2) ∩ PartOf(l1, a) ∩ PartOf(l2, a) ∩ PartOf(b, a) ∩ Attached(l1, b) ∩ Attached(l2, b) …

Measurements

- Units Functions
  - Length(L1) = Inches(1.5) = Centimeters(3.81)
- Conversion
  - Centimeters(2.54 x d) = Inches(d)
- More examples
  - Diameter(Basketballx) = Inches(9.5)
  - ListPrice(Basketballx) = $(19)
  - d E Days => Duration(d) = Hours(24)

Substances and objects

- Individuation
- Count nouns
  - One "cat" cut in two is not two "cats"
  - If it has any **extrinsic** qualities
- Mass nouns
  - One "butter-object" cut in half is two "butter-objects"
  - x E Butter $\cap$ PartOf(y, x) => y E Butter
  - x E Butter => MeltingPoint(x, Centigrade(30))
  - All qualities are **intrinsic**