361CCS

Knowledge-Based Agents

CHAPTER 7, 8

Prepared by Dr. Aliya Aleryani

A knowledge-based agent

- 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**.

Axiom
When the sentence is taken as being given without being derived from other sentences

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 **performs** the chosen **action**.

TAP (Tells, Ask, Perform)

Figure 7.1

function KB-AGENT(*percept*) **returns** an *action*
  persistent: *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*

A generic knowledge-based agent. Given a percept, the agent adds the percept to its knowledge base, asks the knowledge base for the best action, and tells the knowledge base that it has in fact taken that action.

MAKE-PERCEPT-SENTENCE MAKE-ACTION-QUERY MAKE-ACTION SENTENCE

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.

- 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)`

Procedural vs declarative Approaches

- **Declarative Approach of Agent**

Starting with an empty knowledge base, the agent designer can TELL sentences one by one until the agent knows how to operate in its environment

- **Procedural Approach of Agent**

The **procedural** approach encodes desired behaviors directly as program code.

- **The successful agent=Combines of Declarative and Procedural**
- **knowledge-based agent** Learns it self, autonomous

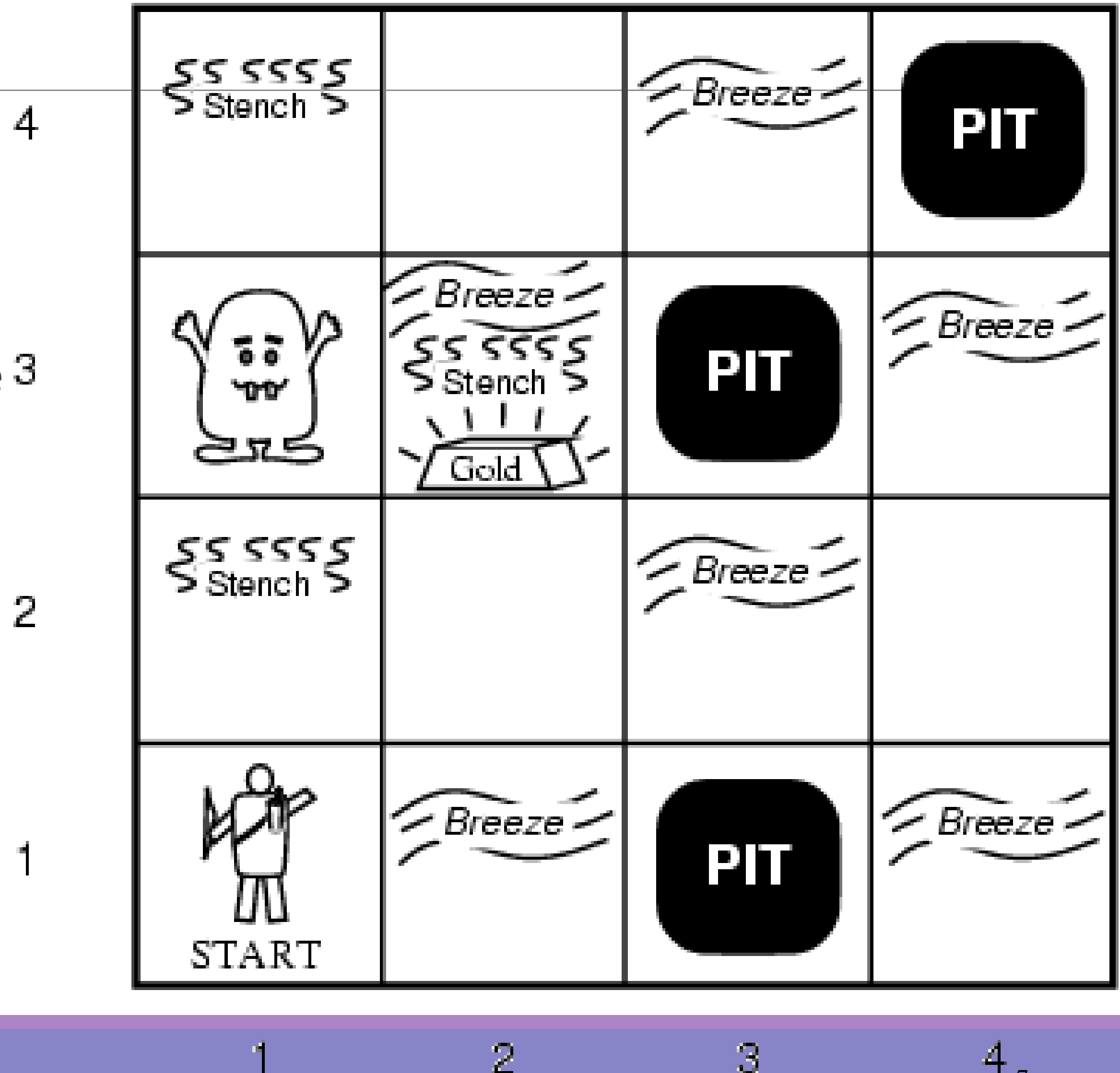
The Wumpus World environment

- 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 without being eaten

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.



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 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]

Wumpus world actions

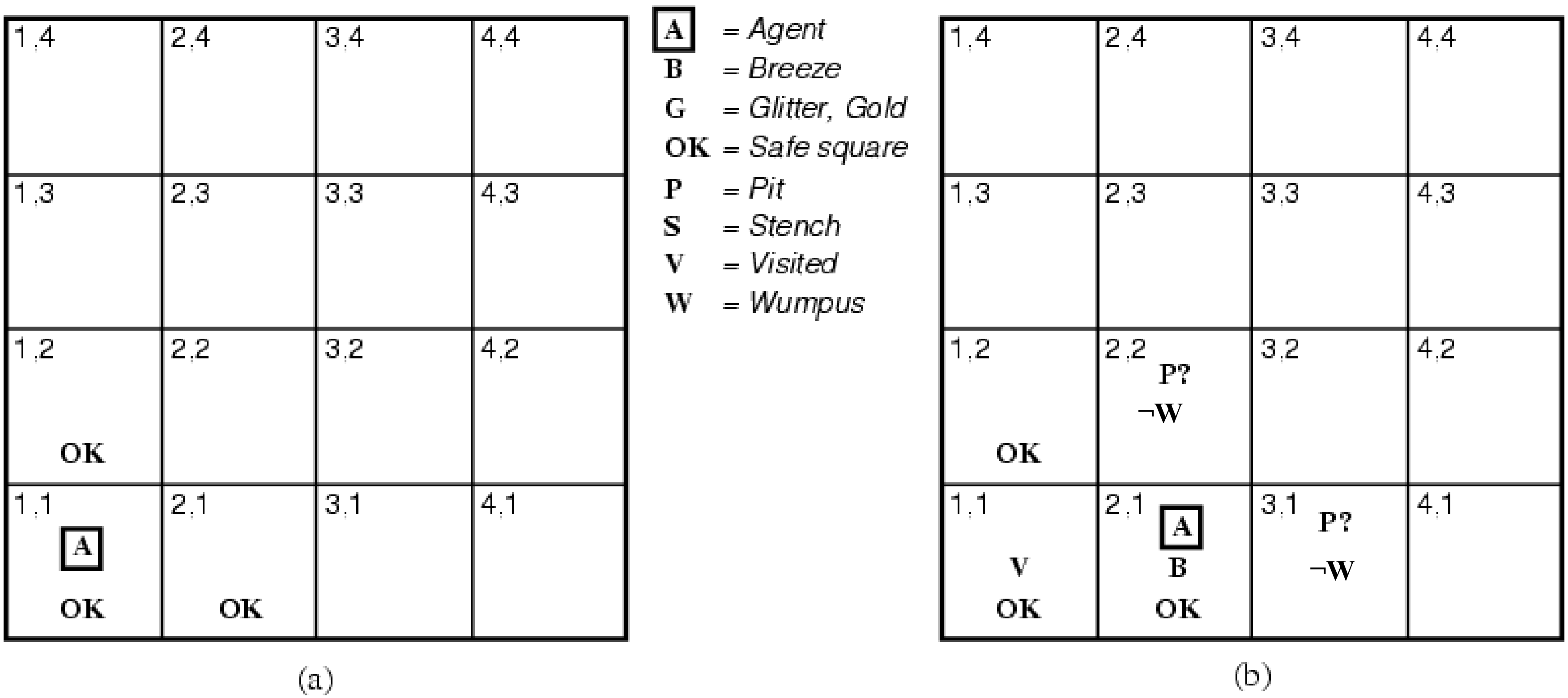
- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab**: Pick up an object that is in the same square as the agent
- **shoot**: Fire an arrow in a straight line in the direction the agent is facing. The arrow continues until it either hits and kills the wumpus or hits the outer wall. The agent has only one arrow, so only the first Shoot action has any effect
- **climb** is used to leave the cave. This action is only effective in the start square
- **die**: This action automatically and irretrievably happens if the agent enters a square with a pit or a live wumpus

Wumpus goal

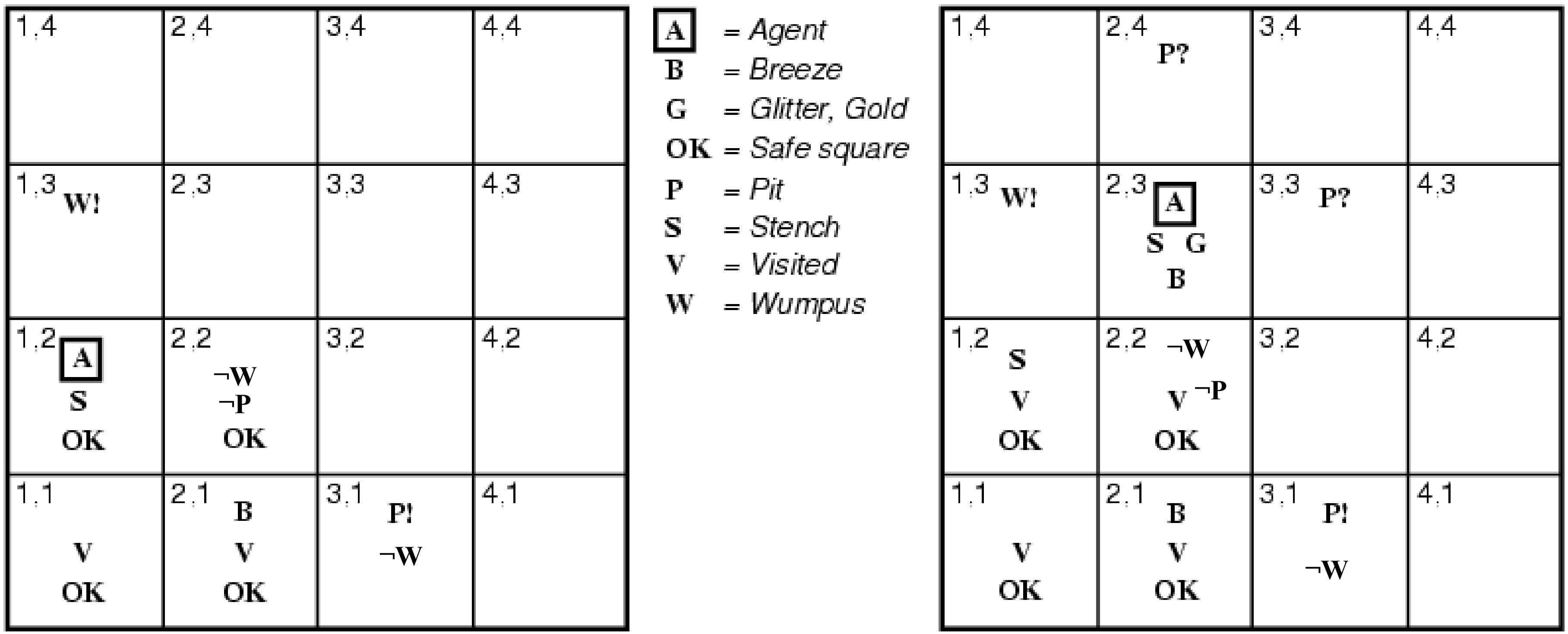
The agent's goal is to find the gold and bring it back to the start square as quickly as possible, without getting killed

- 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

The Wumpus agent’s first step



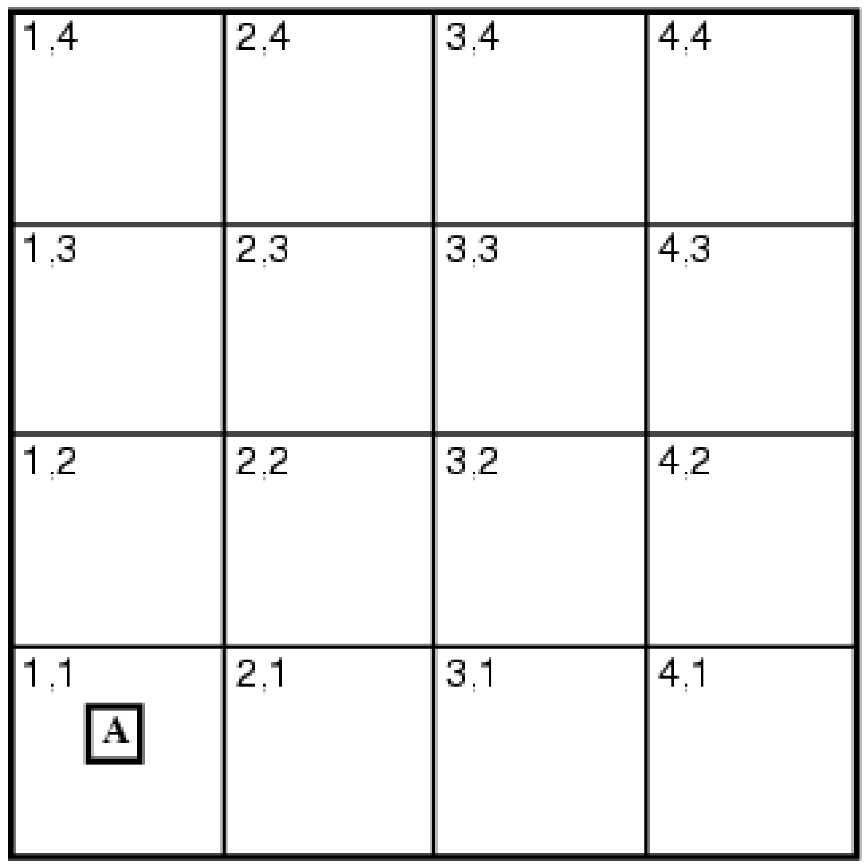
Later



1,4
2,4
3,4
4,4
1,3 W!
2,3
3,3
4,3
1,2 A S OK
2,2 ¬W ¬P OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P! ¬W
4,1
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 ¬W V ¬P OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P! ¬W
4,1

(a) (b)

Let’s Play!



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

Representation, reasoning, and logic

The object of knowledge representation is to express knowledge in a **computer-tractable** form, so that agents can perform well.

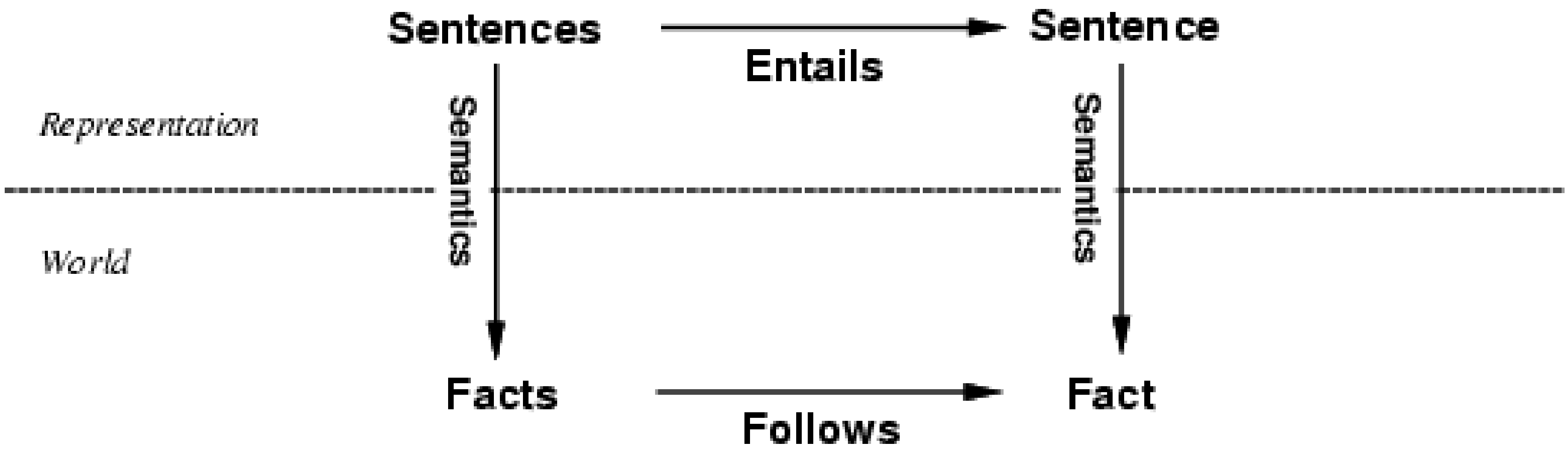
A knowledge representation language is defined by:

- 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.

fundamental property of logical reasoning

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

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.

Fundamental Concepts of logical representation and reasoning

Information is represented in *sentences*, which must have correct *syntax*

(1 + 2) * 7 = 21 vs. 2) + 7 = * (1 21

The semantics of a sentence defines its truth with respect to each possible world – an *interpretation* assigning T or F to all propositions

W is a model of S means that sentence S is true under interpretation W

[What do the following mean?]

- X |= Y
- X entails Y
- Y logically follows from X

Which are true?
Which are not true but useful?

1. {Man, Man → Mortal} |= Mortal
2. {Raining,Dog→ Mammal} |= Mammal
3. {Raining,Raining → Wet} |= Wet
4. {Smoke, Fire → Smoke} |= Fire
5. {Tall ^ Silly} |= Tall
6. {Tall v Silly} |= Silly
7. {Tall, Silly} |= Tall ^ Silly

[Wumpus world EG illustrating possible worlds and entailment]

Entailment (reminder)

A |= B

Under all interpretations in which A is true, B is true as well

All models of A are models of B

Whenever A is true, B is true as well

A entails B

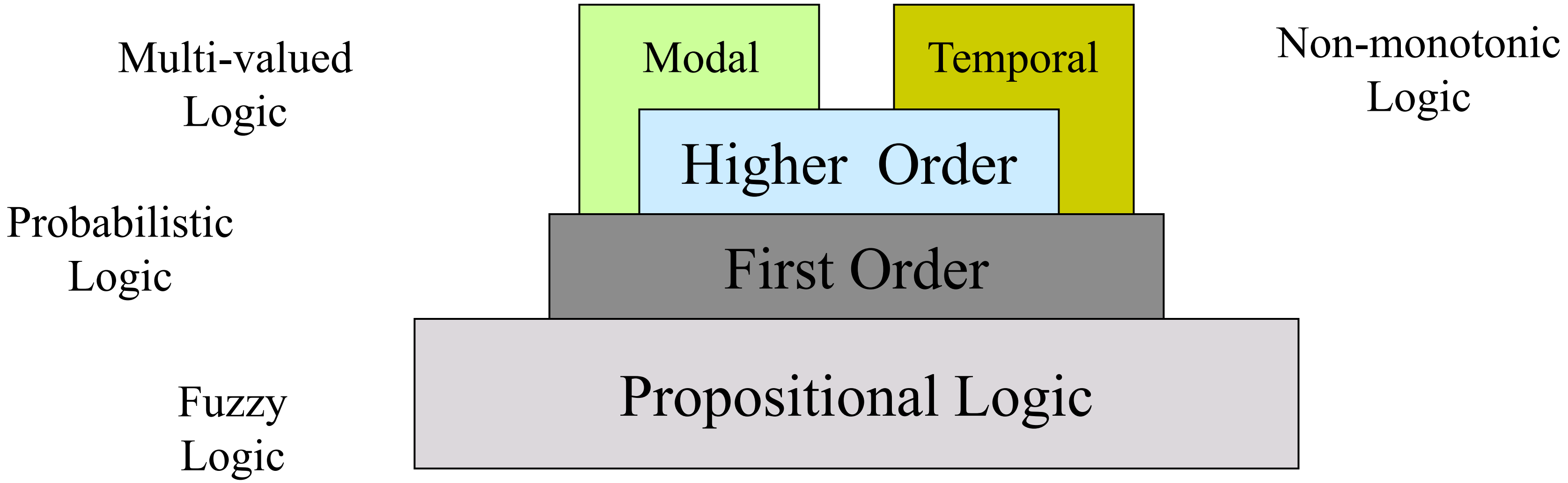
B logically follows from A

Soundness and completeness

A *sound* inference method derives only entailed sentences.

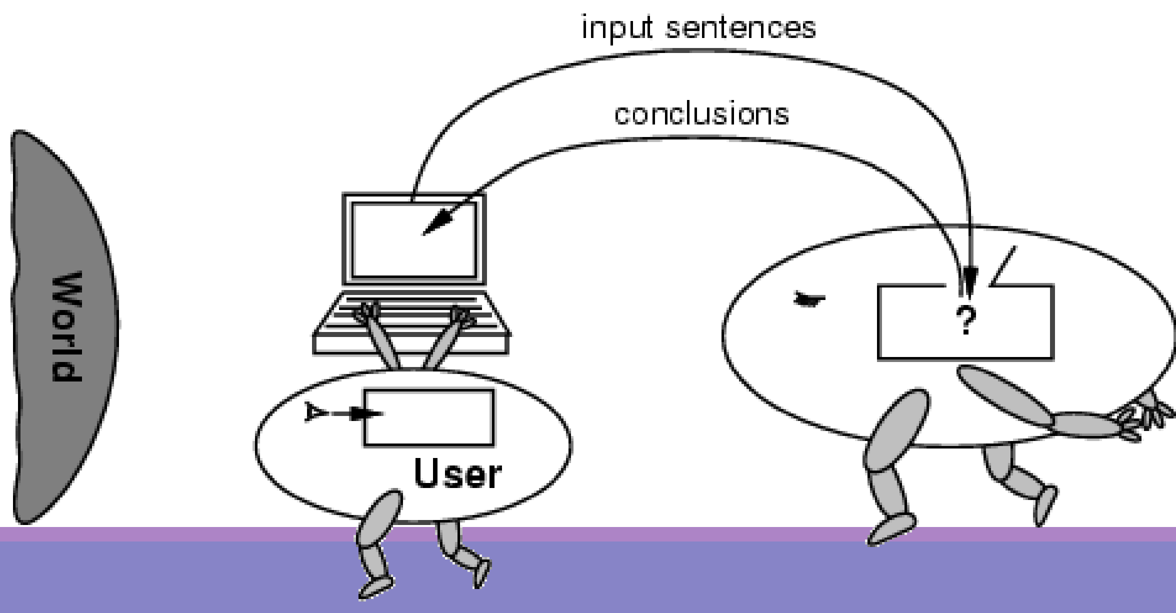
Analogous to the property of *completeness* in search, a *complete* inference method can derive any sentence that is entailed.

Logic as a KR language



No independent access to the world

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



Propositional Logic

Big Ideas

- Logic is a great knowledge representation language for many AI problems
- **Propositional logic** is the simple foundation and fine for some AI problems
- **First order logic** (FOL) is much more expressive as a KR language and more commonly used in AI
- There are many variations: horn logic, higher order logic, three-valued logic, probabilistic logics, etc.

Propositional logic

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

$\wedge$	and	[conjunction]
$\vee$	or	[disjunction]
$\Rightarrow$	implies	[implication / conditional]
$\Leftrightarrow$	is equivalent	[biconditional]
$\neg$	not	[negation]

- **Literal**: atomic sentence or negated atomic sentence
  P, $\neg$ P

Examples of PL sentences

$(P \wedge Q) \rightarrow R$

"If it is hot and humid, then it is raining"

$Q \rightarrow P$

"If it is humid, then it is hot"

Q

"It is humid."

We're free to choose better symbols, btw:

Ho = "It is hot"

Hu = "It is humid"

R = "It is raining"

Propositional logic (PL)

- Simple language for showing key ideas and definitions
- User defines set of propositional symbols, like P and Q
- User defines **semantics** of each propositional symbol:
  - P means "It is hot", Q means "It is humid", etc.
- A sentence (**well formed formula**) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg$S is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then (S $\vee$ T), (S $\wedge$ T), (S $\rightarrow$ T), and (S $\leftrightarrow$ T) are sentences
  - A sentence results from a finite number of applications of the rules

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* – an assignment of truth values to propositional symbols that makes each sentence in the KB True

Model for a KB

Let the KB be [$P \wedge Q \rightarrow R$, $Q \rightarrow P$]

What are the possible models? Consider all possible assignments of T|F to P, Q and R and check truth tables

- **FFF: OK**
- **FFT: OK**
- FTF: NO
- FTT: NO
- **TFF: OK**
- **TFT: OK**
- TTF: NO
- **TTT: OK**

P: it's hot
Q: it's humid
R: it's raining

If KB is [$P \wedge Q \rightarrow R$, $Q \rightarrow P$, Q], then the only model is TTT

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 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.

Tautologies

- A sentence is a tautology if it is true for any setting of its propositional symbols

P	Q	P OR Q	NOT(P) AND NOT(Q)	(P OR Q) OR (NOT(P) AND NOT(Q))
false	false	false	true	true
false	true	true	false	true
true	false	true	false	true
true	true	true	false	true

(P OR Q) OR (NOT(P) AND NOT(Q)) is a tautology

Truth tables

- Truth tables are used to define logical connectives and to determine when a complex sentence is true given the values of the symbols in it.

Truth tables for 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*

Example of a truth table used for 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*

On the implies connective: $P \rightarrow Q$

- Note that $\rightarrow$ is a logical connective

So $P \rightarrow Q$ is a logical sentence and has a truth value, i.e., is either true or false

- If we add this sentence to the KB, it can be used by an inference rule, *Modes Ponens*, to derive/infer/prove Q if P is also in the KB

- Given a KB where P=True and Q=True, we can also derive/infer/prove that $P \rightarrow Q$ is True

$P \rightarrow Q$

When is $P \rightarrow Q$ true? Check all that apply

- ☐ P=Q=true
- ☐ P=Q=false
- ☐ P=true, Q=false
- ☐ P=false, Q=true

$P \rightarrow Q$

When is $P \rightarrow Q$ true? Check all that apply

- [x] P=Q=true
- [x] P=Q=false
- [ ] P=true, Q=false
- [x] P=false, Q=true

We can get this from the truth table for $\rightarrow$

Note: in FOL it's much harder to prove that a conditional true.

- Consider proving prime(x) $\rightarrow$ odd(x)

Inference rules

- **Logical inference** creates new sentences that logically follow from a set of sentences (KB)
- An inference rule is **sound** if every sentence X it produces when operating on a KB logically follows from the KB
  - i.e., inference rule creates no contradictions
- An inference rule is **complete** if it can produce every expression that logically follows from (is entailed by) the KB.
  - Note 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

RULE	PREMISE	CONCLUSION
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A
Double Negation	$\neg\neg A$	A
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$\mathbf{A \vee B, \neg B \vee C}$	$\mathbf{A \vee C}$

Soundness of modus ponens

A	B	$A \rightarrow B$	OK?
True	True	True	√
True	False	False	√
False	True	True	√
False	False	True	√

Resolution

- **Resolution** is a valid inference rule producing a new clause implied by two clauses containing *complementary literals*
  - A literal is an atomic symbol or its negation, i.e., P, ~P
- Amazingly, this is the only interference rule you need to build a sound and complete theorem prover
  - Based on proof by contradiction and usually called resolution refutation
- The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 60s

Resolution

- A KB is actually a set of sentences all of which are true, i.e., a conjunction of sentences.
- To use resolution, put KB into *conjunctive normal form* (CNF), where each sentence written as a disjunction of (one or more) literals
- Example
  - KB: [$P \rightarrow Q$, $Q \rightarrow R \wedge S$]
  - KB in CNF: [$\sim P \vee Q$, $\sim Q \vee R$, $\sim Q \vee S$]
  - Resolve KB(1) and KB(2) producing: $\sim P \vee R$ *(i.e., $P \rightarrow R$)*
  - Resolve KB(1) and KB(3) producing: $\sim P \vee S$ *(i.e., $P \rightarrow S$)*
  - New KB: [$\sim P \vee Q$, $\sim Q \vee \sim R \vee \sim S$, $\sim P \vee R$, $\sim P \vee S$]

Tautologies

$(A \rightarrow B) \leftrightarrow (\sim A \vee B)$

$(A \vee (B \wedge C)) \leftrightarrow (A \vee B) \wedge (A \vee C)$

Soundness of the resolution inference rule

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*

From the rightmost three columns of this truth table, we can see that

$$(\alpha \vee \beta) \wedge (\beta \vee \gamma) \leftrightarrow (\alpha \vee \gamma)$$

is valid (i.e., always true regardless of the truth values assigned to α, β and γ

Proving things

- A **proof** is a sequence of sentences, where each is a premise or is derived from earlier sentences in the proof by an inference rule
- The last sentence is the **theorem** (also called goal or query) that we want to prove
- Example for the "weather problem"

1 Hu	premise	"It's humid"
2 $Hu \rightarrow Ho$	premise	"If it's humid, it's hot"
3 Ho	modus ponens(1,2)	"It's hot"
4 $(Ho \wedge Hu) \rightarrow R$	premise	"If it's hot & humid, it's raining"
5 $Ho \wedge Hu$	and introduction(1,3)	"It's hot and humid"
6 R	modus ponens(4,5)	"It's raining"

Propositional logic: pro and con

Advantages

- Simple KR language sufficient for some problems
- Lays the foundation for higher logics (e.g., FOL)
- Reasoning is decidable, though NP complete, and efficient techniques exist for many problems

Disadvantages

- Not expressive enough for most problems
- Even when it is, it can very “un-concise”

PL is a weak KR 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 (FOL) is expressive enough to represent this kind of information using relations, variables and quantifiers, e.g.,
  - *Every elephant is gray:* $\forall$ x (elephant(x) $\rightarrow$ gray(x))
  - *There is a white alligator:* $\exists$ x (alligator(X) ^ white(X))

First-order logic

First-order logic (FOL) models the world in terms of

- **Objects,** which are things with individual identities
- **Properties** of objects that distinguish them from other objects
- **Relations** that hold among sets of objects
- **Functions,** which are a subset of relations where there is only one "value" for any given "input"

Examples:

- Objects: Students, lectures, companies, cars ...
- Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
- Properties: blue, oval, even, large, ...
- Functions: father-of, best-friend, second-half, one-more-than ...

User provides

Constant symbols, which represent individuals in the world
- Mary
- 3
- Green

Function symbols, which map individuals to individuals
- father-of(Mary) = John
- color-of(Sky) = Blue

Predicate symbols, which map individuals to truth values
- greater(5,3)
- green(Grass)
- color(Grass, Green)

FOL Provides

Variable symbols

- E.g., x, y, foo

Connectives

- Same as in PL: not ($\neg$), and ($\wedge$), or ($\vee$), implies ($\rightarrow$), if and only if (biconditional $\leftrightarrow$)

Quantifiers

- Universal $\forall$**x** or **(Ax)**
- Existential $\exists$**x** or **(Ex)**

Sentences are built from terms and atoms

- A **term** (denoting a real-world individual) is a constant symbol, a variable symbol, or an n-place function of n terms.
  - x and $f(x_1, \ldots, x_n)$ are terms, where each x_i is a term.
  - A term with no variables is a **ground term**
- An **atomic sentence** (which has value true or false) is an n-place predicate of n terms
- A **complex sentence** is formed from atomic sentences connected by the logical connectives:
  - $\neg P$, $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$, $P \leftrightarrow Q$ where P and Q are sentences
- A **quantified sentence** adds quantifiers $\forall$ and $\exists$
- A **well-formed formula (wff)** is a sentence containing no "free" variables. That is, all variables are "bound" by universal or existential quantifiers.
  - $(\forall x)P(x,y)$ has x bound as a universally quantified variable, but y is free.

Quantifiers

Universal quantification

- $(\forall x)P(x)$ means that P holds for **all** values of x in the domain associated with that variable
- E.g., $(\forall x)$ dolphin(x) $\rightarrow$ mammal(x)

Existential quantification

- $(\exists\, x)P(x)$ means that P holds for **some** value of x in the domain associated with that variable
- E.g., $(\exists\, x)$ mammal(x) $\wedge$ lays-eggs(x)
- Permits one to make a statement about some object without naming it

Quantifiers

- Universal quantifiers are often used with "implies" to form "rules":

  $(\forall x)$ student(x) $\rightarrow$ smart(x) means "All students are smart"

- Universal quantification is *rarely* used to make blanket statements about every individual in the world:

  $(\forall x)$student(x)$\wedge$smart(x) means "Everyone in the world is a student and is smart"

- Existential quantifiers are usually used with "and" to specify a list of properties about an individual:

  $(\exists x)$ student(x) $\wedge$ smart(x) means "There is a student who is smart"

- A common mistake is to represent this English sentence as the FOL sentence:

  $(\exists x)$ student(x) $\rightarrow$ smart(x)

  - But what happens when there is a person who is *not* a student?

Some Examples

Brother(Richard, John). Richard is the brother of John

Married(Father(Richard),Mother(John)): Richard the Lionheart's father is married

to King John's mother

∀x King(x) ⇒ Person(x) : All kings are persons

∀x ∀y Brother(x,y) ⇒ Sibling(x,y) Brothers are siblings

∀x ∃ y Loves(x,y) Everybody loves somebody

∀x ¬Likes(x,Parsnips) = ¬∃ x Likes(x,Parsnips)

∀x Likes(x,IceCream) is equivalent to ¬∃ x ¬Likes(x,IceCream)

The De Morgan rules for quantified and unquantified sentences

Prove the following by using Truth table:

$$
\begin{array}{llll}
\neg\exists x \; P \equiv \forall x \; \neg P & \qquad & \neg(P \vee Q) \equiv \neg P \wedge \neg Q \\
\neg\forall x \; P \equiv \exists x \; \neg P & & \neg(P \wedge Q) \equiv \neg P \vee \neg Q \\
\forall x \; P \equiv \neg\exists x \; \neg P & & P \wedge Q \equiv \neg(\neg P \wedge \neg Q) \\
\exists x \; P \equiv \neg\forall x \; \neg P & & P \vee Q \equiv \neg(\neg P \wedge \neg Q) .
\end{array}
$$

Propositional logic summary

- Inference is the process of deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives 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
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - Simple syntax and semantics suffices to illustrate the process of inference
  - Propositional logic can become impractical, even for very small worlds

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.