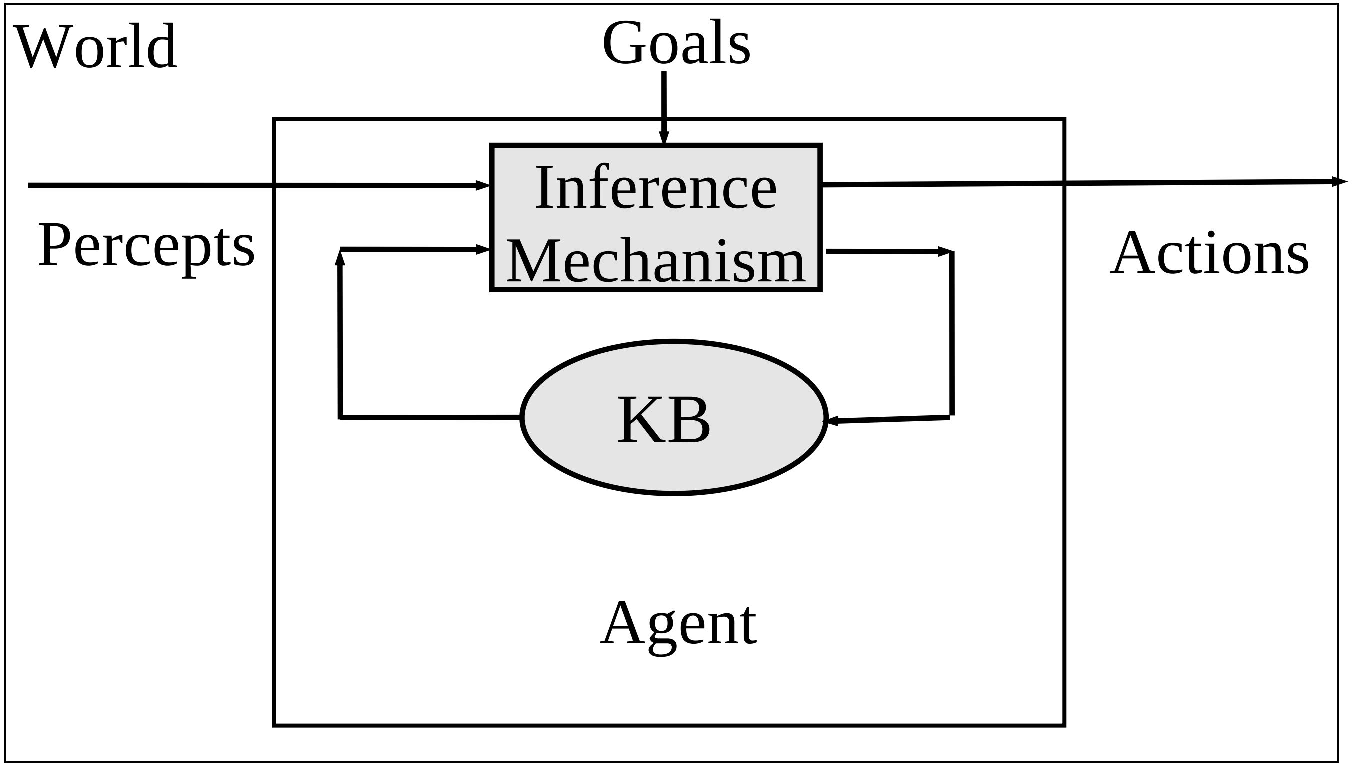
Agents That Reason Logically



A Knowledge-Based Agent

Knowledge Base (KB)

- The Central component of a knowledge-based agent.
- A set of **facts** about the world represented in **sentences,** which are expressed in the **knowledge representation language**.
- The **KB** initially starts off with some background knowledge.

Inference Mechanism

- Another main component of a knowledge-based agent.
- To determine what follows from what the KB has been TELLed.

Characteristics

- The agent **TELL**s the knowledge base what it perceives before **ASK**s the knowledge base what action it should perform.
- The answer to an **ASK**ed question should follow from what has been **TELL**ed to the KB previously.

```
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
```

Figure 1: A generic knowledge-based agent.

✧The functions MAKE-PERCEPT-SENTENCE and MAKE-ACTION-QUERY hide the detail of the representation language.

✧A knowledge-based agent can be described at 3 levels:

1. **knowledge** or **epistemological level** - describe the agent by what it knows.

   e.g. A direct flight from Taipei to Beijing.

2. **logical level** (i.e. sentences) - knowledge is encoded into sentences.

   e.g. flight(direct, Taipei, Beijing).

3. **implementation level** - physical representations of sentences at the logical level.

   e.g. (list flight direct Taipei Beijing).

🗁 Building a Knowledge-Based Agent

- ✧It is possible to construct a knowledge-based agent by TELLing it what it needs to know.
- ✧The **background knowledge** of an agent represent the designer‘s knowledge of the environment.
- ✧The **declarative approach** is the desirable method for building systems.
- ✧A **learning mechanism** can be built into an agent to make it fully autonomous.

The Wumpus World Environment

A computer game to explore the relationship between a knowledge-based agent and the environment.

Specifying the Environment

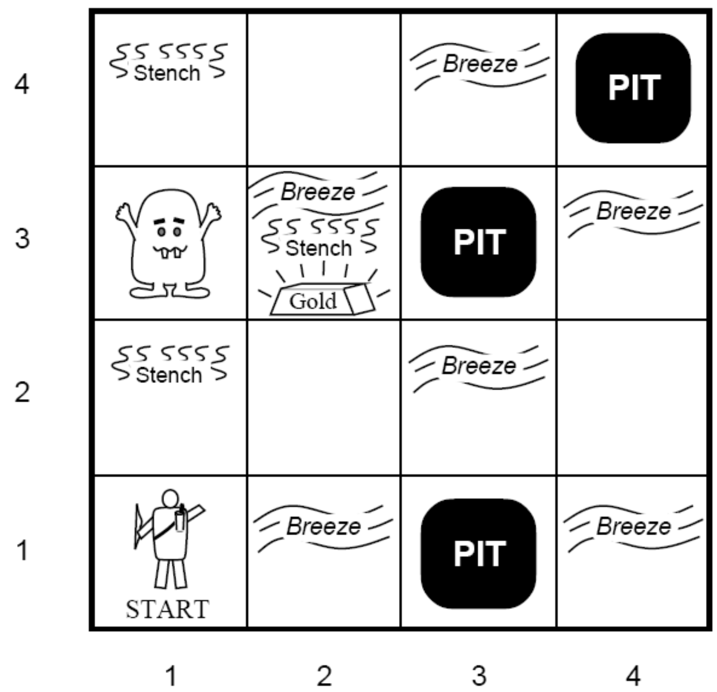
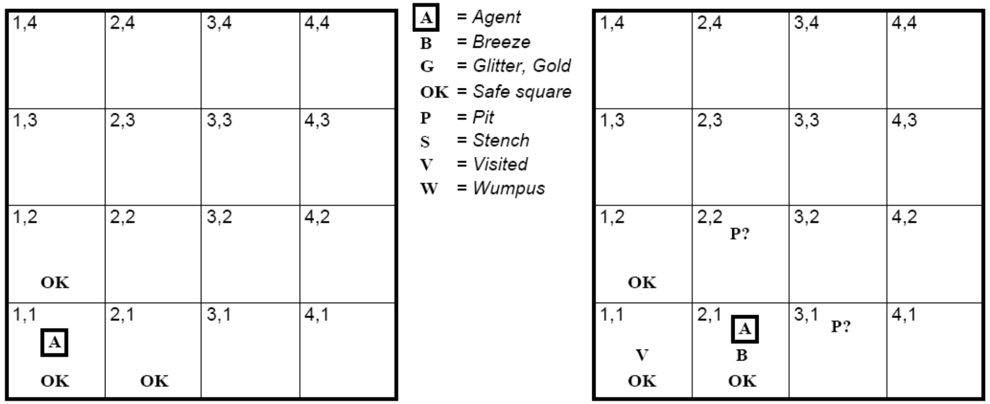


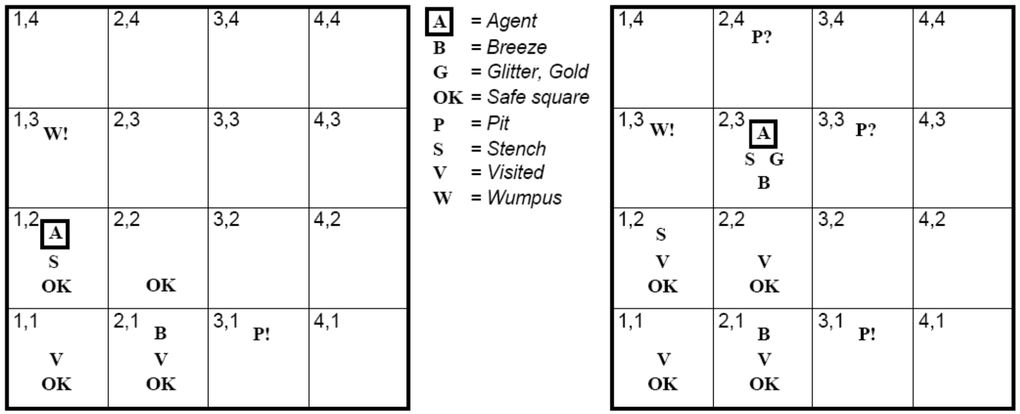
Figure 2: A typical wumpus world.

🗁 Acting and Reasoning in the Wumpus World



(a) (b)

Figure 3: The first step taken by the agent in the wumpus world. (a) The initial situation, after percept [*None, None, None, None, None*]. (b) After one move, with percept [*None, Breeze, None, None, None*].



(a) (b)

Figure 4: Two later stages in the progress of the agent. (a) After the third move, with percept [*Stench, None, None, None, None*]. (b) After the fifth move, with percept [*Stench, Breeze, Glitter, None, None*].

✧The agent needs a fairly sophisticate inference mechanism, because it must combine knowledge gained at different times in different places.

Representation, Reasoning, and Logic

The operation of a knowledge-based agent is supported by a **representation language** and its **reasoning mechanism**; they both are, in turn, based on some form of **logic**.

The **logic** of a representation language may be defined from its **syntax** and **semantics**.

Reasoning is a process of constructing new physical configurations (sentences) from old ones.

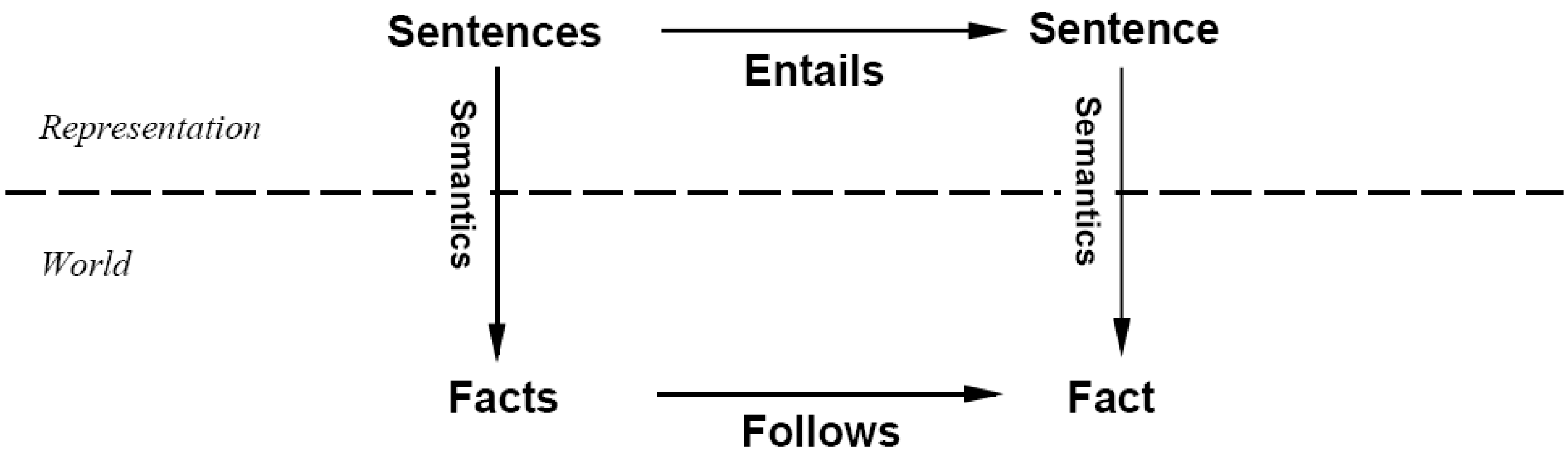


Figure 5: The connection between sentences and facts is provided by the semantics of the language. The property of one fact following from some other facts is mirrored by the property of one sentence being entailed by some other sentences. Logical inference generates new sentences that are entailed by existing sentences.

"**Proper reasoning**" should ensure that manipulations on internal symbols via syntax (sentences in an agent) must retain the correct semantics of represented truth (facts of the world).

Improper reasoning example:

FIRST VILLAGER: We have found a witch. May we burn her?
ALL: A witch! Burn her!
BEDEVERE: Why do you think she is a witch?
SECOND VILLAGER: She turned *me* into a newt.
BEDEVERE: A newt?
SECOND VILLAGER *(after looking at himself for some time)*: I got better.
ALL: Burn her anyway.
BEDEVERE: Quiet! Quiet! There are ways of telling whether she is a witch.
BEDEVERE: Tell me . . . what do you do with witches?
ALL: Burn them.
BEDEVERE: And what do you burn, apart from witches?
FOURTH VILLAGER: . . . Wood?
BEDEVERE: So why do witches burn?
SECOND VILLAGER: *(pianissimo)* Because they're made of wood?
BEDEVERE: Good.
ALL: I see. Yes, of course.
BEDEVERE: So how can we tell if she is made of wood?
FIRST VILLAGER: Make a bridge out of her.
BEDEVERE: Ah . . . but can you not also make bridges out of stone?
ALL: Yes, of course . . . um . . . er . . .
BEDEVERE: Does wood sink in water?
ALL: No, no, it floats. Throw her in the pond.
BEDEVERE: Wait. Wait . . . tell me, what also floats on water?
ALL: Bread? No, no no. Apples . . . gravy . . . very small rocks . . .
BEDEVERE: No, no no,
KING ARTHUR: A duck!
(They all turn and look at ARTHUR. BEDEVERE looks up very impressed.)
BEDEVERE: Exactly. So . . . logically . . .
FIRST VILLAGER *(beginning to pick up the thread)*: If she . . . weighs the same as a duck . . . she's made of wood.
BEDEVERE: And therefore?
ALL: A witch!

Figure 6.6 An example of "logical" reasoning gone wrong. (Excerpted with permission from *Monty Python and the Holy Grail*, © 1977, Reed Consumer Books.)

Entailment: given an initial set of sentences that are **true**, newly generated sentences must be true also.

Entailment relation between a KB and a sentence can be expressed in a mathematical notation as:

"KB $\models \alpha$" reads as "KB entails α"

An **inference procedure** can be described by the sentences that it can derive. In a logician notation:

"KB $\vdash_i \alpha$" reads "α is derived from **KB** by **i**"

An inference procedure that generates only entailed sentences is called **Sound** or **truth-preserving**.

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

An inference procedure is **complete** if it can find a proof for any sentence that is entailed.

*For a **sound inference**, the inference steps, given a knowledge base KB, should only derive new sentences that represent facts that follow from the facts represented by the KB.*

Logical languages deal with more or less *everything* we might want to represent and about which we might want to reason.

By examining the semantics of logical languages we can extract the **proof theory** of the language, which specifies the reasoning steps that are sound.

For example:

$$E=MC^2 \quad \Rightarrow \quad ET = MC^2T$$

🗁 Representation

✧A good knowledge representation scheme should have the following desired characteristics:

1. expressive -- “there is a pit in [2,2] or [3,1]”
2. concise -- declarative.
3. unambiguous -- “small dogs and cats”
4. context independent -- what is said today should still be interpretable tomorrow.

(NOTE: 1 and 2 conflict with 3 and 4)

- ✧The person who wrote a sentence should provide an **interpretation** for it, i.e. a mapping of **symbols** to **meanings**.
- ✧A sentence is **true** under a particular interpretation if the state of affairs it represents is the case.
- ✧In a **compositional** language, the meaning of a sentence is a function of the meaning of its parts.

🗁 Inference

Inference or **reasoning** is any process by which conclusions are reached.

✧**logical inference** or **deduction** is a ***sound*** reasoning process which generates only entailed sentences.

Validity and **satisfiability**

✧ **Valid** or **necessarily TRUE sentences** are true in all interpretations in all possible worlds. they are also called **analytic sentences** or **tautologies**.

E.g. “there is a stench at [1,1] or there is not a stench at [1,1]”

✧ A sentence is **satisfiable** if and only if there is some interpretation in some world for which it is true.

E.g. “there is a wumpus at [1,2]”

Inference in computer

✧ Two handicaps of computers:

1. It does not necessarily know the interpretation you are using for the sentences in the KB.
2. It knows nothing at all about the world except what appears in the KB.

✧ It is the **inference procedure** that has to show that a sentence "*If KB is true then A is true*" is a valid sentence.

✧ A **formal inference mechanism** can derive valid conclusions even the computer does not know the interpretation you are using.

✧ What makes formal inference powerful is that there is no limit to the complexity of the sentences it can handle.

🗁 Logics

A logic consists of the following:

✧A formal system for describing states of affairs, consisting of

(a) the **syntax** of the language, which describes how to make sentences, and

(b) the **semantics** of the language, which states the systematic constraints on how sentences relate to states of affairs.

✧The **proof theory --** a set of rules for deducing the entailments of a set of sentences.

✧In **propositional logic**:

1. Symbols represent whole propositions (facts), and
2. **Boolean connectives** are used to combine propositions to generate sentences with more complex meaning.

✧In **First Order logic**:

1. Use **objects** and **predicates on objects** (properties of objects or relations between objects) to represent the states of worlds.
2. **Connectives** and **quantifiers** can be used to write sentences about everything in the universe at once.

✧**Ontological commitments**: have to do with the nature of reality, i.e. pertain to facts outside the agent.

✧**Epistemological commitments**: have to do with the possible states of knowledge an agent can have using various type of logic, i.e. pertain to knowledge representation inside agent.

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

Figure 7: Formal languages and their ontological and epistemological commitments.

Propositional Logic: A Very Simple Logic

Severs to illustrate many of the concepts of Logic

Syntax

- logical constants ***True*** and ***False***
- A symbol is a sentence, e.g P or Q.
- wrapped sentence (P$\wedge$Q) is also a sentence.
- logical connectives:

$\wedge$(and) P$\wedge$Q is a **conjunction** with **conjuncts** P and Q.
$\vee$(or) P$\vee$Q is a **disjunction** with **disjuncts** P and Q.
$\Rightarrow$(imply) P$\Rightarrow$Q is called an **implication** (or **condition**) with **premise** or **antecedent** P and **conclusion** or **consequent** Q.
$\Leftrightarrow$(equivalent) P$\Leftrightarrow$Q is an **equivalence** (or **bi-conditional**).
$\neg$(not) P is the **negation** of P.

✧ BNF grammar in (***Figure 8***) introduces **atomic sentences**, which consist of a single symbol(e.g P), and **complex sentences**, which contains connectives or parenthese(e.g., P∧Q).

Propositional logic syntax

Sentence	→ *AtomicSentence* | *ComplexSentence*
AtomicSentence	→ True | False | *Symbol*
Symbol	→ P | Q | R | . . .
ComplexSentence	→ ¬*Sentence*
	| (*Sentence* ∧ *Sentence*)
	| (*Sentence* ∨ *Sentence*)
	| (*Sentence* ⇒ *Sentence*)
	| (*Sentence* ⇔ *Sentence*)

BNF grammar in propositional logic

$((\neg P \vee ((\mathrm{True} \wedge R) \Leftrightarrow Q)) \Rightarrow S$ well formed

$(\neg(P \vee Q) \wedge \Rightarrow S)$ not well formed

- ✧Atomic sentences or negated atomic sentences are also called **literals**.
- ✧Ambiguous sentences are parsed according to the order of precedence $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$.

$$\neg P \vee Q \wedge R \Rightarrow S \quad \equiv \quad ((\neg P) \vee (Q \wedge R)) \Rightarrow S.$$

🗁 Semantics

- ✧A sentence containing just a propositional symbol is satisfiable but not valid; it is true just when the fact it refers to is the case.
- ✧A complex sentence has a meaning derived from the meaning of its parts. **Truth table** is used to define the 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*

Figure 9: Truth tables for the five logical connectives.

- ✧Propositional logic does not necessarily imply any **causal relation** among atomic sentences, e.g. “5 is odd implies Tokyo is the capital of Japan” is **true**!!

🗁 Validity and Inference

✧Truth table is used for testing the valid of sentences.

✧Inference process is to construct a truth table for a sentence ***Premise*** $\Rightarrow$ ***Conclusion*** to prove that the ***Conclusion*** is entailed by the ***Premise***.

✧For example:

P : there is a wumpus in [1,3].
H : there is a wumpus in [2,2].
at some point we learn that **(P∨H)** and **¬H**, then we can conclude that P is true, i.e. there is a wumpus in [1,3], via **((P∨H) ∧ ¬H) ⇒ 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*

Figure 10: Truth table showing validity of a complex sentence.

✧It is essential that a reasoning system be able to draw conclusions that follow from the premises, regardless of the world to which the sentences are intended to refer.

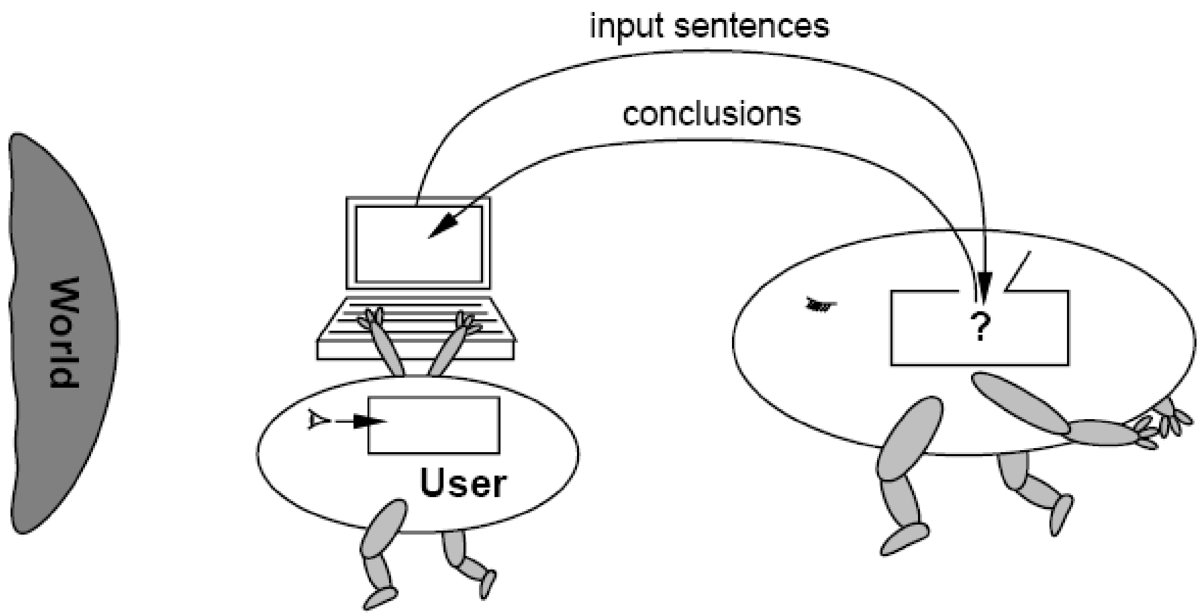


Figure 11: Sentences often refer to a world to which the agent has no independent access.

Models

✧Any world in which a sentence is true under a particular interpretation is called a **model** of that sentence under that interpretation.

✧For example, the world in Figure 2 is a model of sentence $S_{1,2}$ under the interpretation that it means **" there is a stench in [1,2]"**.

✧Weaker claim about the world results more models of a same sentence. ***The more conjunctions are added into the knowledge base, the fewer the models there will be***.

✧Defing **entailment** by models -- If models of a KB are all models of a sentence α then KB **entails** α. That is, whenever the KB is true, α must also be true.

✧Defining the meaning of a sentence by means of set operations on sets of models -- **(*Figure 12*)**.

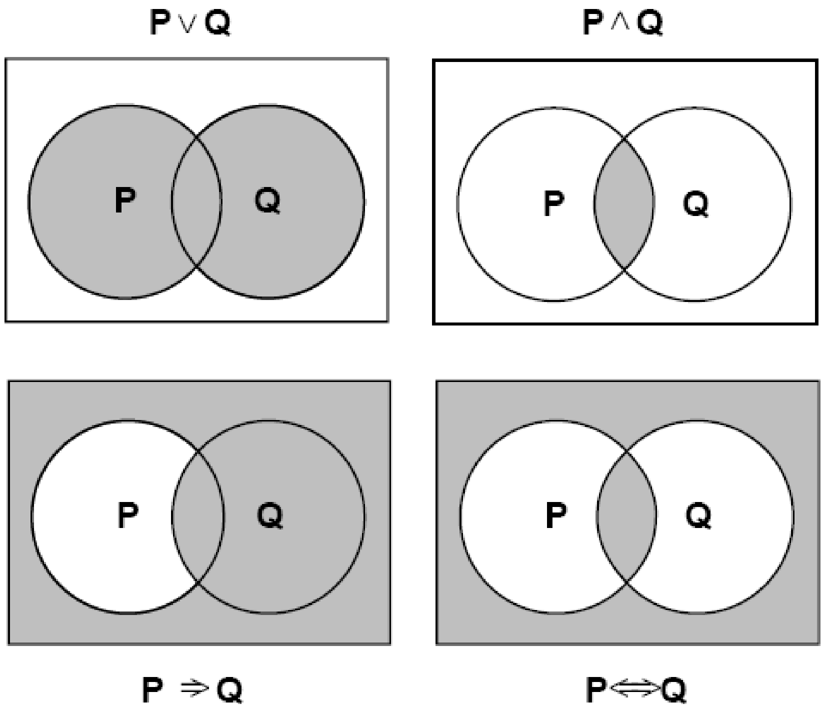


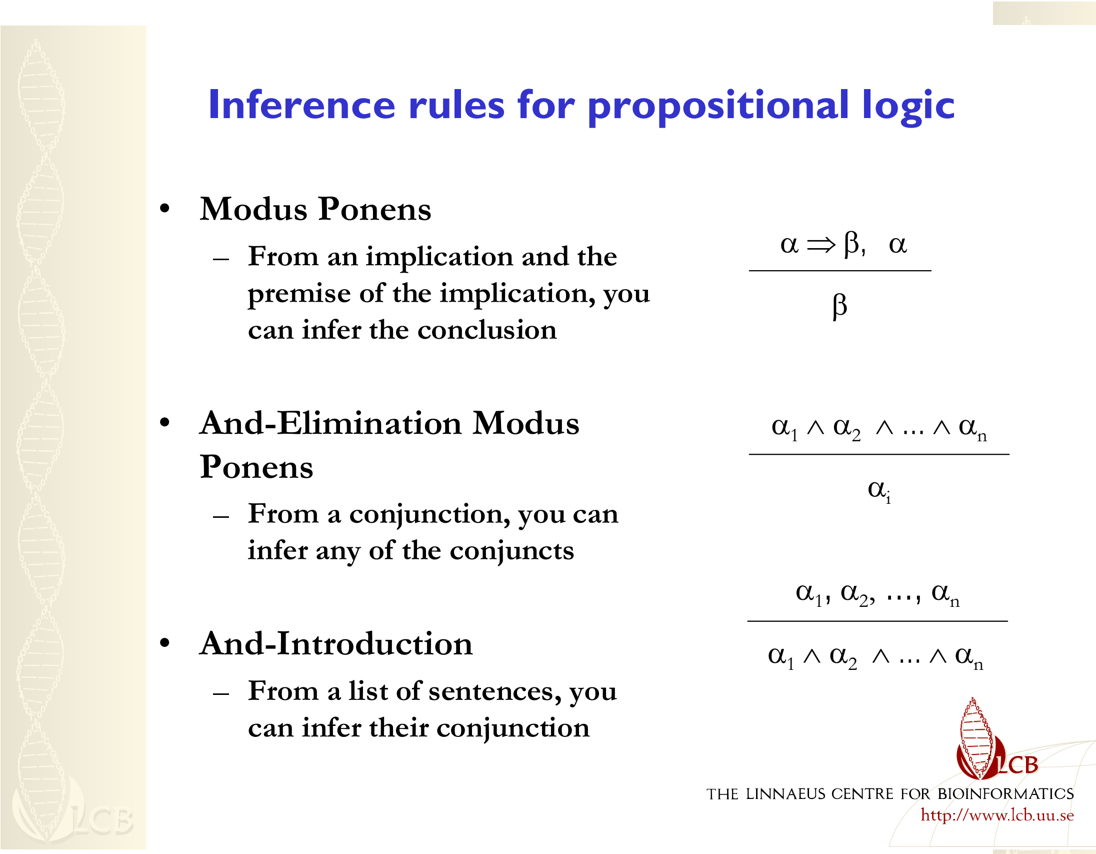
Figure 12: Models of complex sentences in terms of the models of their components. In each diagram, the shaded parts correspond to the model of the complex sentence.

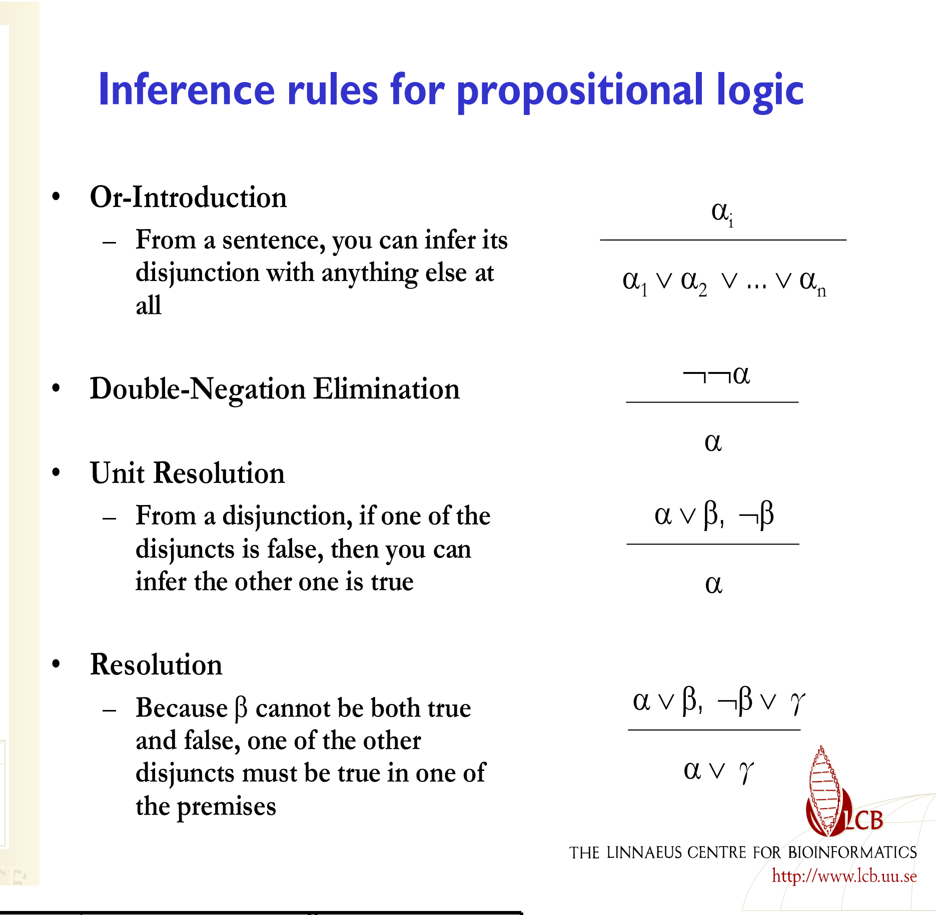
Rules of Inference for Propositional Logic

Inference rules captures the *patterns of inferences* that are proved **sound**.

Using $\frac{\alpha}{\beta}$ to denote an inference rule -- whenever something in the KB maches the pattern above the line, the inference rule concludes the premise below the line.

For Propositional Logic, there are seven commonly used inference :





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>

Figure 14: A truth table demonstrating the soundness of the resolution inference rule. We have underlined the rows where both premises are true.

🗁 Complexity of Propositional Inference

✧Inference in propositional logic:

✓Truth-table method is complete with $\mathbf{2^n}$ entries.

✓Checking a set of sentences for satisfiability is NP-complete.

✧Monotonicity (Performance may be improved)

A logic is ***monotonic*** if when after adding some new sentences to the KB, all the sentences entailed by the original KB are still entailed by the new larger KB.

$$\textbf{if} \quad KB_1 \models \alpha \quad \textbf{then} \quad (KB_1 \cup KB_2) \models \alpha$$

Monotonicity makes inference rules **local** because their premises need only be compared with a small portion of the

Propositional and First-order logic are monotonic.

✧**Horn Clauses**(Horn Sentences)

A useful class of sentences for which polynomial-time inference procedure exists.

They have format as:

$\mathbf{P_1 \wedge P_2 \wedge \ldots P_n \Rightarrow Q}$ (all literals are non-negated)

$\cong \quad \mathbf{\neg P_1 \vee \neg P_2 \vee \ldots \neg P_n \vee Q}$

An Agent for the Wumpus World

Show a snapshot of a propositional logic agent reasoning about the wumpus world.

Assume at the point of:

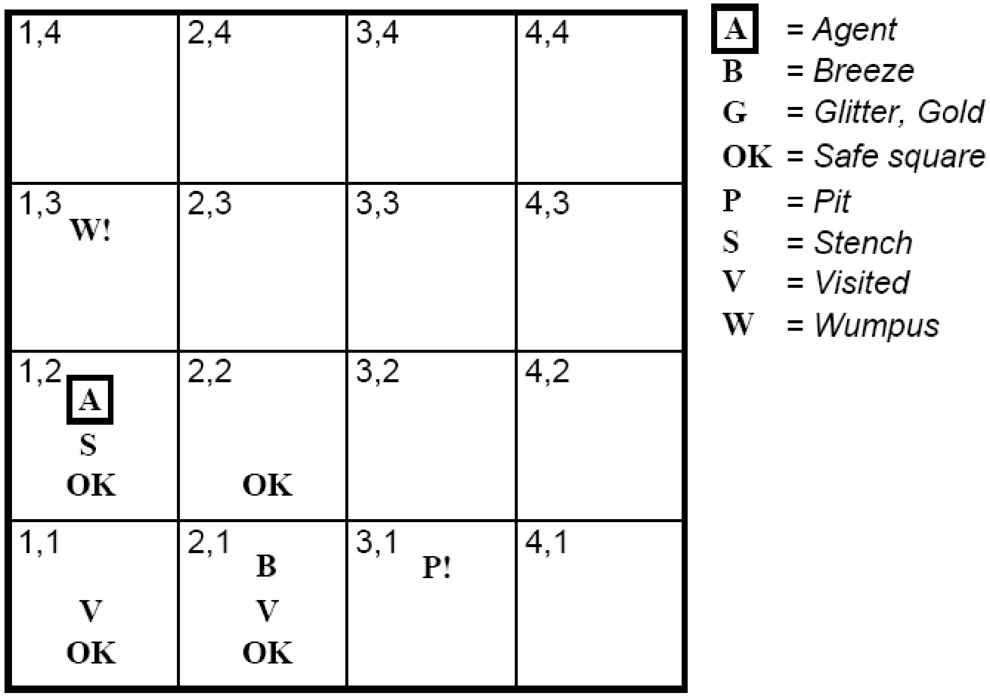


Figure 15: The agent's knowledge after the third move. The current percept is [*Stench*, *None*, *None*, *None*, *None*].

🗁 The Knowledge Base

✧For the moment of Figure 15, the KB contains:

$\neg \mathbf{S_{1,1}}$: there is no stench in [1,1]
$\neg \mathbf{S_{2,1}}$: there is no stench in [2,1]
$\mathbf{S_{1,2}}$: there is a stench in [1,2]
$\neg \mathbf{B_{1,1}}$: there is no breeze in [1,1]
$\mathbf{B_{2,1}}$: there is a breeze in [2,1]
$\neg \mathbf{B_{1,2}}$: there is no breeze in [1,2]

✧Background knowledge: (partial)

$$\mathbf{R_1} : \neg \mathbf{S_{1,1}} \Rightarrow \neg \mathbf{W_{1,1}} \wedge \neg \mathbf{W_{1,2}} \wedge \neg \mathbf{W_{2,1}}$$
$$\mathbf{R_2} : \neg \mathbf{S_{2,1}} \Rightarrow \neg \mathbf{W_{1,1}} \wedge \neg \mathbf{W_{2,1}} \wedge \neg \mathbf{W_{2,2}} \wedge \neg \mathbf{W_{3,1}}$$
$$\mathbf{R_3} : \neg \mathbf{S_{1,2}} \Rightarrow \neg \mathbf{W_{1,1}} \wedge \neg \mathbf{W_{1,2}} \wedge \neg \mathbf{W_{2,2}} \wedge \neg \mathbf{W_{1,3}}$$
$$\mathbf{R_4} : \mathbf{S_{1,2}} \Rightarrow \mathbf{W_{1,3}} \vee \mathbf{W_{1,2}} \vee \mathbf{W_{2,2}} \vee \mathbf{W_{1,1}}$$

Finding the Wumpus

To prove $\mathbf{KB \Rightarrow W_{1,3}}$ by using inference rules instead of constructing truth table for these 12 symbols, which produce $2^{12}=4096$ rows.

1. Apply Modus Ponens with $\neg S_{1,1}$ and the sentence R_1 to get :

   $\neg \mathbf{W_{1,1}} \wedge \neg \mathbf{W_{1,2}} \wedge \neg \mathbf{W_{2,1}}$

2. Apply And-Elimination to 1. to get:

   $\neg \mathbf{W_{1,1}} \quad \neg \mathbf{W_{1,2}} \quad \neg \mathbf{W_{2,1}}$

3. Apply Modus Ponens to $\neg S_{2,1}$ and R_2, and then And-Elimination to the result to get :

   $\neg \mathbf{W_{2,1}} \quad \neg \mathbf{W_{2,2}} \quad \neg \mathbf{W_{3,1}}$

4. Apply Modus Ponens to $S_{1,2}$ and R_4, and then And-Elimination to the result to get :

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

5. Apply the unit resolution rule, where α is $W_{1,3} \vee W_{1,2} \vee W_{2,2}$ and β is $W_{1,1}$ ($\neg W_{1,1}$ was derived in step 2) to get :

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

6. Apply unit resolution again with $W_{1,3} \vee W_{1,2}$ as α and $W_{2,2}$ as β ($\neg W_{2,2}$ was derived in step 3) to get :

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

7. Finally, one more unit resolution with $W_{1,3}$ as α and $W_{1,2}$ as β ($\neg W_{1,2}$ was derived in step 2) to get :

$\mathbf{W_{1,3}}$ **The wumpus is in [1,3]**

🗁 Translating Knowledge into Action

Rules must also be specified for the agent for each location and orientation, such as:

$$\mathbf{A_{1,1} \wedge East_{A} \wedge W_{2,1} \Rightarrow \neg \mathit{Forward}}$$

✧Propositional logic is not able to answer questions such as; "What action should I take?".

✧It is only possible to answer questions such as "Should I go forward?" or "Should I turn right?"

✧A possible agent with propositional logic is as in

```
function PROPOSITIONAL-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))
   for each action in the list of possible actions do
      if ASK(KB, MAKE-ACTION-QUERY(t, action)) then
         t ← t + 1
         return action
   end
```

Figure 16: A knowledge-based agent using propositional logic.

Problems with Propositional Agent

✧Too many propositions to handle.

"*Don't go forward if the wumpus is in front of you*" is represented by a set of 64 rules (16 squares ×4 orientations).

✧Hard to deal with changes of the world.

$$\left.\begin{array}{l}
\mathbf{A}^{0}_{1,1} \wedge \mathbf{East}^{0}_{A} \wedge \mathbf{W}^{0}_{2,1} \Rightarrow \neg \mathbf{Forward}^{\mathbf{0}} \\
\mathbf{A}^{1}_{1,1} \wedge \mathbf{East}^{1}_{A} \wedge \mathbf{W}^{1}_{2,1} \Rightarrow \neg \mathbf{Forward}^{\mathbf{1}} \\
\mathbf{A}^{2}_{1,1} \wedge \mathbf{East}^{2}_{A} \wedge \mathbf{W}^{2}_{2,1} \Rightarrow \neg \mathbf{Forward}^{\mathbf{2}} \\
\quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\mathbf{A}^{2}_{1,1} \wedge \mathbf{North}^{2}_{A} \wedge \mathbf{W}^{2}_{1,2} \Rightarrow \neg \mathbf{Forward}^{\mathbf{2}} \\
\quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots
\end{array}\right\} \textbf{6400} \text{ rules for 100 time step}$$