COSC1125/1127 Artificial Intelligence

Week 4: Knowledge Representation I (Propositional Logic)

[RN2] Sec 7.1-7.6 Chapters 8-9
[RN3] Sec 7.1-7.6 Chapters 8-9

Knowledge and Reasoning

A good chess program can defeat human masters. However it does not know that a chess board can be used for playing checker as well. It cannot decide when to play a chess game.

Humans have knowledge about things and can do reasoning. It is also important for an artificial being, we call it ***agent***, to achieve "high level" intelligence.

Knowledge and reasoning

- enable agents to cope with complex environments.

  [Eg. how to schedule an around-world trip? Need the knowledge about "the world".]

- play a crucial role in dealing with partially observable environments.

  [Part of states could be hidden. Eg. how to avoid collisions in a busy shopping mall?]

- enable agents to handle complicated tasks, eg. understanding natural language.

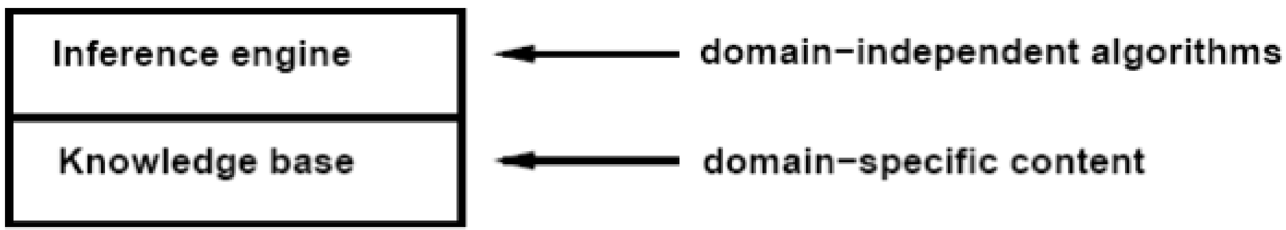
  [Eg."Mike opened the door, found a chair and sat on it". It refers to what? door or chair?]

- provide more flexibility in problem solving.

  [Decision based on updated knowledge. Eg. how to choose a good text book for a course?]

Knowledge and Reasoning…

To be intelligent, an agent needs a ***knowledge base*** to store knowledge and an ***inference engine*** to do the reasoning.



An agent must be able to:

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

Next, we discuss how to represent knowledge and how to perform inference.

Knowledge Representation

It is desirable to have a Knowledge Representation (KR) scheme with the following properties:

- **Expressive:** must be able to represent as much as possible about the world.
- **Precise:** must be clear and unambiguous.

  An example of ambiguity: *The man saw the boy in the park with the telescope.*

  Whose telescope? the man's or the boy's?

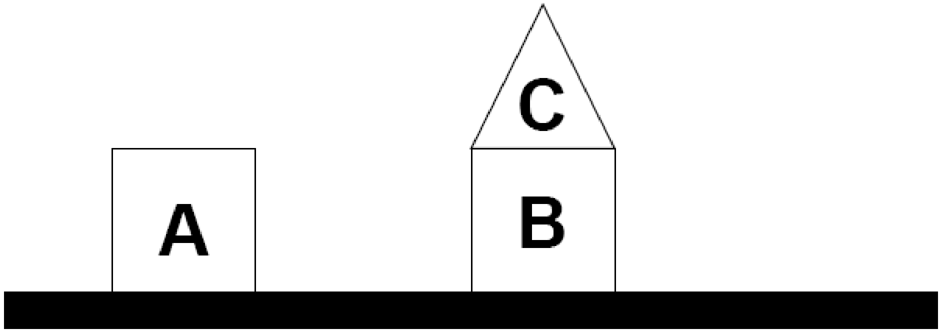
- **Regular:** must be a clear mapping between the knowledge and its representation.
- **Adaptable**: must be able to add new information or delete invalid information.
- **Suitable for reasoning**: new knowledge can be inferred
- *more on next slide…*

Knowledge Representation…

More desirable properties of a Knowledge Representation (KR) scheme:

- **Computationally attractive:** the scheme is able to be implemented.

- **Qualitative:** it can represent qualitative knowledge. For example



How to represent “ you can only put block A on block B if B is clear” ?

- **Meta-level reasoning:** can represent knowledge about knowledge.

- Others: inheritance, knowledge of uncertainty, etc……

Physical Symbol Hypothesis

The following are ***necessary*** and ***sufficient*** conditions for intelligent behaviour:

1. Symbolic patterns for the representation of significant aspects of a problem domain.

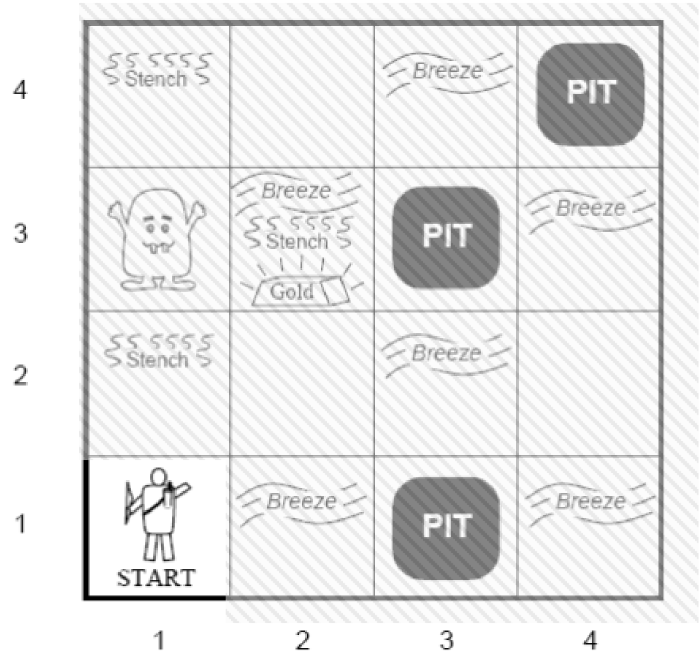
2. Operations or processes on these patterns which are used to generate new patterns which are potential solution to problems.

3. Search to select a solution from among these possibilities.

The physical symbol hypothesis is the foundation of all work in (Symbolic) Artificial Intelligence.

Example: Wumpus World

The hunter must find the gold without being eaten by wumpus or falling in a pit.



In this world

- Squares adjacent to wumpus are smelly
- Squares adjacent to pit are breezy
- Glitter if gold is in the same square

The hunter can only perceive one square.

Wumpus World

To solve the wumpus world problem, an “intelligent” hunter should:

1. use symbolic patterns to represent the information about wumpus world. Such as “squares adjacent to wumpus are smelly”, “I am in a breezy square” etc.

2. define operations which can be used to search a solution. Such as “move left”, “move up”, etc. Other actions include “grab” (an object) and “shoot” (fire an arrow).

3. perform a search to find the solution, in this case, reaching gold.

[Compare this slide with the “Physical Symbol Hypothesis” slide.]

Wumpus World

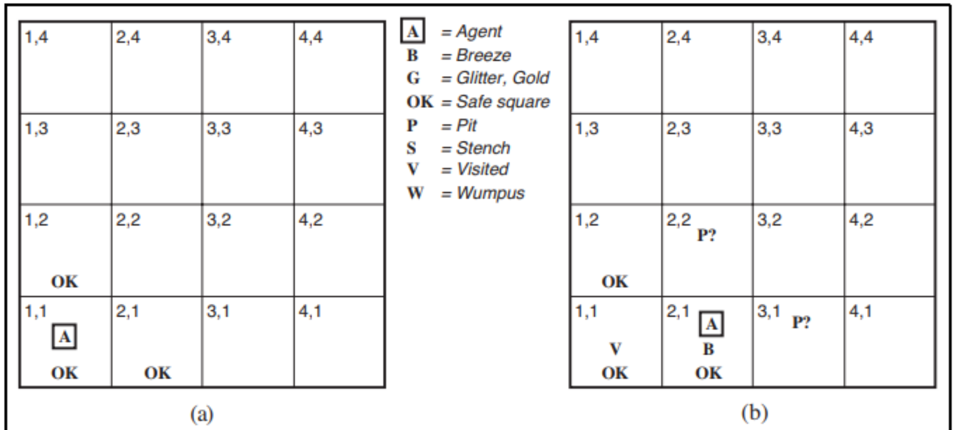


Figure 7.3 The first step taken by the agent in the wumpus world. (a) The initial situation, after percept [*None*, *None*, *None*, *None*, *None*]. (b) After one move, with percept [*None*, *Breeze*, *None*, *None*, *None*].

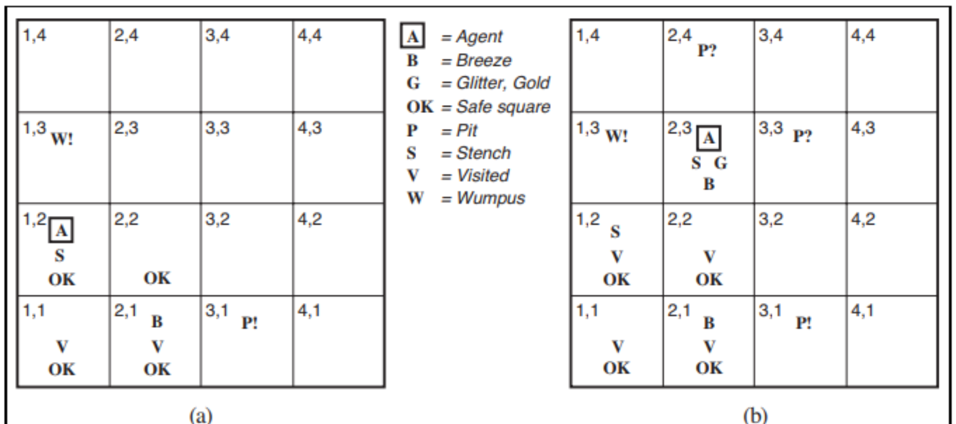


Figure 7.4 Two later stages in the progress of the agent. (a) After the third move, with percept [*Stench*, *None*, *None*, *None*, *None*]. (b) After the fifth move, with percept [*Stench*, *Breeze*, *Glitter*, *None*, *None*].

The agent has 5 sensors, which provide information for the percept:

- In the square containing the Wumpus and in the directly adjacent squares the agent will perceive a stench.
- In the square directly adjacent to a pit, the agent will perceive a breeze.
- In the square where the gold is, the agent will perceive a glitter.
- When an agent walks into a wall, it will perceive a bump.
- When the Wumpus is killed, it emits a woeful scream that can be perceived anywhere in the cave.

For example, if there is a stench and breeze, but no glitter, bump, or scream, the agent will receive the percept: [Stench, Breeze, None, None, None]

Wumpus World…

Wumpus world is partially observable (only local perception).

Some rules of the game in the Wumpus world:

> *If there is a breeze in square (X , Y), then there is a pit in one of the adjacent squares.*
>
> *If there is a glitter at a square, then gold is detected.*
>
> *If there is a stench, then the wumpus is in an adjacent square.*

If the agent can make use of the information collected from the start of the game, and make inferences using known rules, then the agent should be able to **draw a conclusion** from the available information. The conclusion is guaranteed to be correct if the available information is correct.

This is fundamental property of **logical** reasoning.

Logic

Logic is the science of reasoning, proof, thinking and inference – *Concise Oxford English Dictionary.*

It is an important formalism for knowledge representation.

It is a powerful inferencing method.

Logic is a set of formal languages for representing information such that conclusion can be drawn.

For example, "**α is true so β is true**" or "**β is true where α is true**" is represented as

$$\alpha \models \beta$$

This expression is called **entailment**, one thing follows logically from another. α entails β. If α is true, then β must also be true.

Symbol α could mean "X > Y", β could mean " Y < X". Then the above expression means "if X >Y then Y< X". Sounds logical ?

Logic

There are three main components of a logic system:

1. **Syntax:** defines legal expressions, sentences in the language.

   Use the language of arithmetic as analog:
   $x + 2 > 3$ is a legal sentence, but $x + 5 - <$ is not.

   A grammatically correct sentence in logic is called **Well Formed Formula (WFF).**

2. **Semantics:** defines the "meaning" of sentences – whether they hold the truth.

   Use the language of arithmetic as analog:
   $x + 2 > x$ is always true.
   $3 - 1 = 0$ is always false.

3. **Inference mechanism:** for deciding what follows validly from a set of logical sentences.

Next, we will study a simple form of logic – propositional logic.

Propositional Logic

Propositional logic (PL) is the simplest logical system.

The basic symbols of PL are:

- **logical constants** (*T* or *True*, *F* or *False*)
- **Proposition symbols** (e.g. P, Q, R, …)
- **Connectives** ($\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$)

Connectives are used to build more complex sentences from simpler ones.

For example, if $P_{3,1}$ stands for "*There is a pit in square 3,1*" and $P_{3,2}$ stands for "*There is a pit in square 3,2*", then

$P_{3,1} \vee P_{3,2}$ stands for "*There is a pit in square 3,1* ***or*** *square 3,2*"

$P_{3,1} \wedge P_{3,2}$ stands for "*There is a pit in square 3,1* ***and*** *square 3,2*"

Sentences in Propositional Logic

The logical constants *True* and *False* are sentences.

If P is a sentence then (P) is a sentence.

If P and Q are sentences then $P \wedge Q$ is a sentence (**conjunction**).

If P and Q are sentences then $P \vee Q$ is a sentence (**disjunction**).

If P and Q are sentences then $\neg P$ is a sentence (**negation**).

If P and Q are sentences then $P \Rightarrow Q$ is a sentence (**implication**).

If P and Q are sentences then $P \Leftrightarrow Q$ is a sentence (**biconditional**).

A **literal** is an atomic sentence, either positive P or negative $\neg P$.

Examples of PL Sentence

The following sentences are WFFs (Well Formed Formula):

P

$P \wedge Q$

$P \vee Q$

$(P \vee Q) \Rightarrow R$

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

Note: Parentheses must be used to avoid ambiguity.

The following sentences are **not** WFFs

$P \neg$

$P \wedge \neg$

$\wedge P Q$

$\Rightarrow P$

Examples of PL Sentence...

Question: are the following sentences WFFs?

$P \wedge P$

$P \vee (\neg Q)$

$P \vee \neg Q$

$\neg \neg Q$

$\neg \neg \neg \neg Q$

$((\neg \neg P) \Rightarrow (Q \vee (S \wedge (R \Rightarrow T))))$

True $\Rightarrow$ *False*

We now know the **syntax** aspects of propositional logic. Next, we will look at the **semantics** aspects of PL.

Semantics

Semantics define the "meaning" of sentences.

The meaning of a logical sentence is based on whether it is true or false.

- Truth values of **proposition symbols** are assigned by **interpretations**, which correspond to the way the world could be, such as symbols

$P_{3,1}$	refers to	*There is a pit in square 3,1* (wumpus world)
Q	refers to	$x > 3$

[Depends on the situation, the above symbols could be true or false.]

- Truth values of **WFF**s are worked out from truth value of components.

Interpretations

Whether a proposition symbol is true? It depends on its interpretation. Is P true? It depends on what P represents.

What is the truth value of P if it stands for

- *Squares adjacent to wumpus are smelly*
- *Elephants can fly.*
- *It is raining.*
- $X + 5 > X$
- $5 = 2$
- $X + Y > 0$

Some propositions are always true. Some are always false. Some could be true or false, depend on its current state.

Also, propositions are interpreted within a certain scope or a world. "*Squares adjacent to wumpus are smelly*" might not be true outside the wumpus world. "*Elephants can fly*" might be true in a fantasy world.

Truth Value of Sentences

True value of a sentence is based on the truth values of it proposition symbols.

$P \wedge Q$ is true, if and only if (iff) P is true **and** Q is true.

$P \vee Q$ is true, iff either P is true **or** Q is true.

$\neg P$ is true, iff P is false.

$P \Rightarrow Q$ is true, if P is false, **or** both P and Q are true.

$P \Leftrightarrow Q$ is true, iff P and Q are both true or P and Q are both false.

Use a true table to summarize the truth value of these simple sentences:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Implication

The truth values of implication $\Rightarrow$ is somewhat count-intuitive.

Let P refers to "It is raining", Q refers to "I have an umbrella".
So $P \Rightarrow Q$ means "If it is raining then I have an umbrella".

Is this statement true then?

- True if P true and Q true – *If it is raining then I have an umbrella*.
- False if P true and Q false – *If it is raining then I do not have an umbrella*.
- **True** if P **false** and Q **false** – *If it is not raining then I do not have an umbrella*.
- **True** if P **false** and Q **true** – *If it is not raining then I have an umbrella*.

"*5 is odd implies Tokyo is the capital of Japan*" sounds odd but logically true (i.e, it does not require any relation of causation or relevance between *P* and *Q*).

Any implication is true whenever its premise/antecedent is false, e.g., "*5 is even implies Sam is smart*".

If *P* is true, then I am claiming that *Q* is true, otherwise I am making no claim.

Truth Table

For a complex sentence, truth table can be used to determine the truth value by determining truth values of its components.

For example $(P \wedge Q) \Rightarrow (R \vee Q)$ which is always true.

P	Q	R	$P \wedge Q$	$R \vee Q$	$(P \wedge Q) \Rightarrow (R \vee Q)$
T	T	T	T	T	T
T	T	F	T	T	T
T	F	T	F	T	T
T	F	F	F	F	T
F	T	T	F	T	T
F	T	F	F	T	T
F	F	T	F	T	T
F	F	F	F	F	T

Question: how many rows are there in a true table if it has ***n*** proposition symbols?

Wumpus World Sentences

Now we are ready to represent information of the wumpus world in PL.

Let $P_{i,j}$ be true if there is a pit in square [i, j].
Let $B_{i,j}$ be true if there is a breeze in square [i, j].

$\neg P_{1,1}$	There is no pit in square [1, 1].
$\neg B_{1,1}$	There is no breeze in square [1, 1].
$B_{2,1}$	There is a breeze in square [2, 1].
$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$	A square is breezy iif an adjacent square has a pit.
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$	A square is breezy iif an adjacent square has a pit.

Based on these given information, we can infer that there is no pit in square [1,2] !!

How? we will study some key concepts of logic before revealing the inference process.

Equivalence

Two sentences are logically equivalent if and only if (iff) they are true in the same models (they have the same truth value for same input). It is expressed as:

$$\alpha \equiv \beta \text{ iff } \alpha \models \beta \text{ and } \beta \models \alpha$$

Some standard logical equivalences:

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) \quad \text{de Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) \quad \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

$P \Rightarrow Q$ is logically equivalent to $\neg P \lor Q$.

Validity

A sentence is **valid** if it is true in **all** models.

The following sentences are valid:

True

$P \vee \neg P$

$P \Rightarrow P$

$(P \wedge (P \Rightarrow Q)) \Rightarrow Q$

Valid sentences are also known as **tautologies**.

Validity can be used for inference – **Deduction Theorem**:

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

Satisfiability

A sentence is **satisfiable** if it is true in **some** models e.g.

$P \vee Q$,

P

A sentence is **unsatisfiable** if it is true in **no** models e.g.

$P \wedge \neg P$

Satisfiability can be used for inference as:

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

It is actually the standard mathematical proof technique of *reductio ad absurdum* (literally, “reduction to an absurd thing”).

(It is also called proof by refutation or proof by contradiction.)

Inference

The aim of logical inference is to decide whether an entailment $\alpha \models \beta$ is true.
To prove it, there are two main approaches:

- by true table enumeration: compare their truth values of all possible inputs.

  The complexity of this approach is exponential. Why?

- by applying inference rules

Inference rule

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

This general form of inference rules means " *if we have proof that each α_i is true then this is extended to a proof that β is true*".

A chained inference (i.e. possibly using more than one rule) from α to β (α infers β) is written:

$$\alpha \vdash \beta$$

Inference Rules

One well known inference rule is called **Modus Ponens**:

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

It means that if $\alpha \Rightarrow \beta$ and α are given, then β can be inferred.
For example if (*Hungry* $\wedge$ *FoodisAvailable*) $\Rightarrow$ *Eat* and *Hungry* $\wedge$ *FoodisAvailable* are given, then *Eat* can be inferred.

Another useful inference rule is **And-Elimination**:

$$\frac{\alpha \wedge \beta}{\alpha}$$

It means from a conjunction, any of the conjuncts can be inferred.
For example, from (*Hungry* $\wedge$ *FoodisAvailable*), *Hungry* can be inferred.

Inference Rules…

Logical equivalence can be used as inferences rules, for example biconditional elimination yields two rules:

$$\frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta} \qquad \frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}$$

By using these rules, we can infer that there is no pit in [1,2] (the wumpus world).

We already know that:

1: $\neg P_{1,1}$ 2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$ 3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$ 4: $\neg B_{1,1}$ 5: $B_{2,1}$

Apply biconditional elimination to sentence 2, then we can obtain:

6: $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

Inference Rules…

Apply And-Elimination $\frac{\alpha \wedge \beta}{\alpha}$ to sentence 6, we can infer that:

$7: ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

Apply contraposition to sentence 7, we get:

$8: (\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1}))$

Apply Modus Ponens $\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$ to sentence 8 (with Sentence 4 $\neg B_{1,1}$) , we get

$9: \neg (P_{1,2} \vee P_{2,1})$

Apply de Morgan's rule to sentence 9, we can infer that:

$10: \neg P_{1,2} \wedge \neg P_{2,1}$

So there is not pit in square [1,2] and [2,1].

Monotonicity

The previous inference process did not use sentence 1, 3, 5

1: $\neg P_{1,1}$ 2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$ 3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$ 4: $\neg B_{1,1}$ 5: $B_{2,1}$

because they don't have relevant propositions.

Finding a proof can be highly efficient in practice because it can ignore irrelevant propositions, no matter how many of them there are.

This property of logical systems follows a fundamental property: **Monotonicity.**

Monotonicity means for any sentences α and β

$$\text{if } KB \models \alpha \text{ then } KB \wedge \beta \models \alpha$$

Additional information β cannot invalidate any conclusion α already inferred.

Conclusion follow inference rules regardless of what else is in the knowledge base.

Horn Clauses

Inference could be done through the *forward chaining* and *backward chaining*. Before studying that, we need know **Horn Clauses** and **AND-OR graph**.

A Horn clause is a disjunction of literals (atomic sentences) of which ***at most one is positive***.

For example, the clause ($\neg L_{1,1} \lor \neg Breeze \lor B_{1,1}$), where $L_{1,1}$ means that the agent's location is [1,1], is a horn clause.

Horn clauses can be written as an implication whose premise is a conjunction of positive literals and whose conclusion is a single positive literal.

$\neg L_{1,1} \lor \neg Breeze \lor B_{1,1}$ can be written as: $L_{1,1} \land Breeze \Rightarrow B_{1,1}$

(It says that the agent is in [1,1] and there is a breeze, then [1,1] is breezy.)

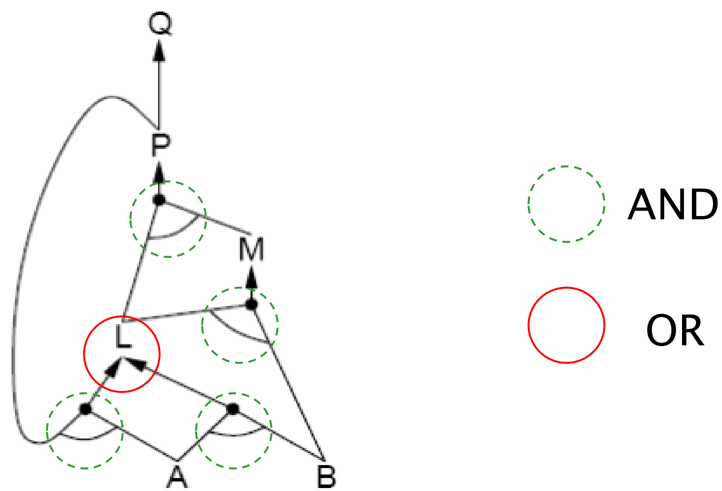
Real-world knowledge bases often only contain Horn clauses.

AND-OR Graph

Horn clauses can be converted to implications and represented as **AND-OR graph.**

The following simple knowledge base of Horn clauses can be represented as:

$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



In AND-OR graphs,

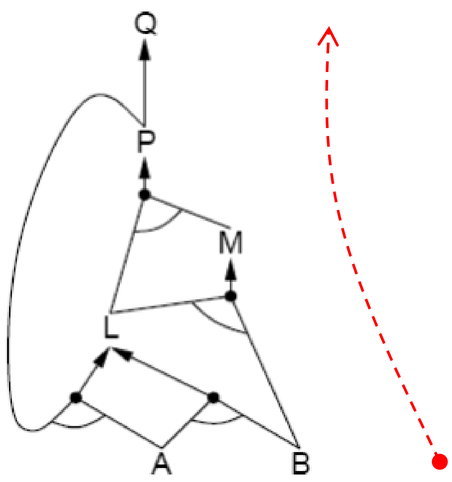
-multiple links joined by an arc indicate a **conjunction** – every link must be proved.

-multiple links without an arc indicate a **disjunction** – any link can be proved.

Forward Chaining

Forward chaining (FC) is data-driven reasoning – starts with the known data.

The basic idea is: fire any rule whose premises are satisfied in the knowledge base until the conclusion is reached.



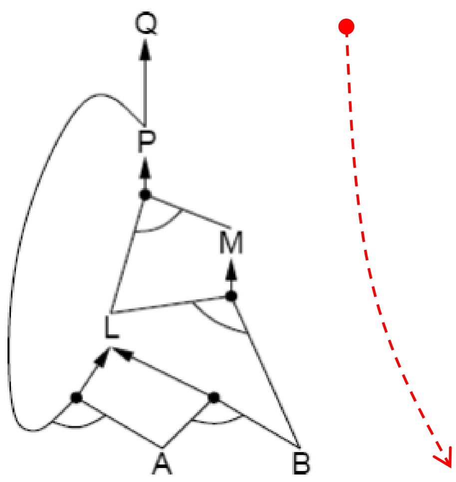
In the example, A and B are known. The FC inference starts from A, B.

Forward chaining is complete, every entailed atomic sentence will be derived.

Backward Chaining

Backward chaining (BC) is a kind of goal-given reasoning – starts with the goal.

The basic idea is: work backwards from the goal q. To prove q, check if q is known or prove all premises of some rule which concludes q (sub-goals).



In the example, Q is the goal. The BC inference starts from Q.

To avoid loops, we need check whether a new sub-goal has already been reached.

Limitations of Propositional Logic

Consider the following facts:

- *The wumpus is dead.*
- *The wumpus is smelly.*

In PC, these must be represented as different propositional symbols. There is no indication that these assertions about the *same* object.

Consider the following facts:

- *The wumpus is dead.*
- *Elvis is dead.*

In PC, these must be represented as different propositional symbols as well. There is no indication that these are the *same* property of different individuals.

Acknowledgement: the slides were developed based on notes from Russell & Norvig, by several computer science staff members over the years.