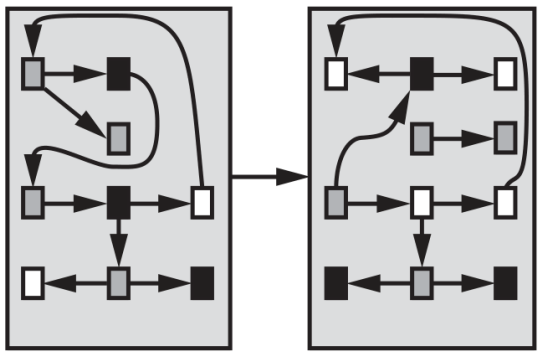
CS761 Artificial Intelligence

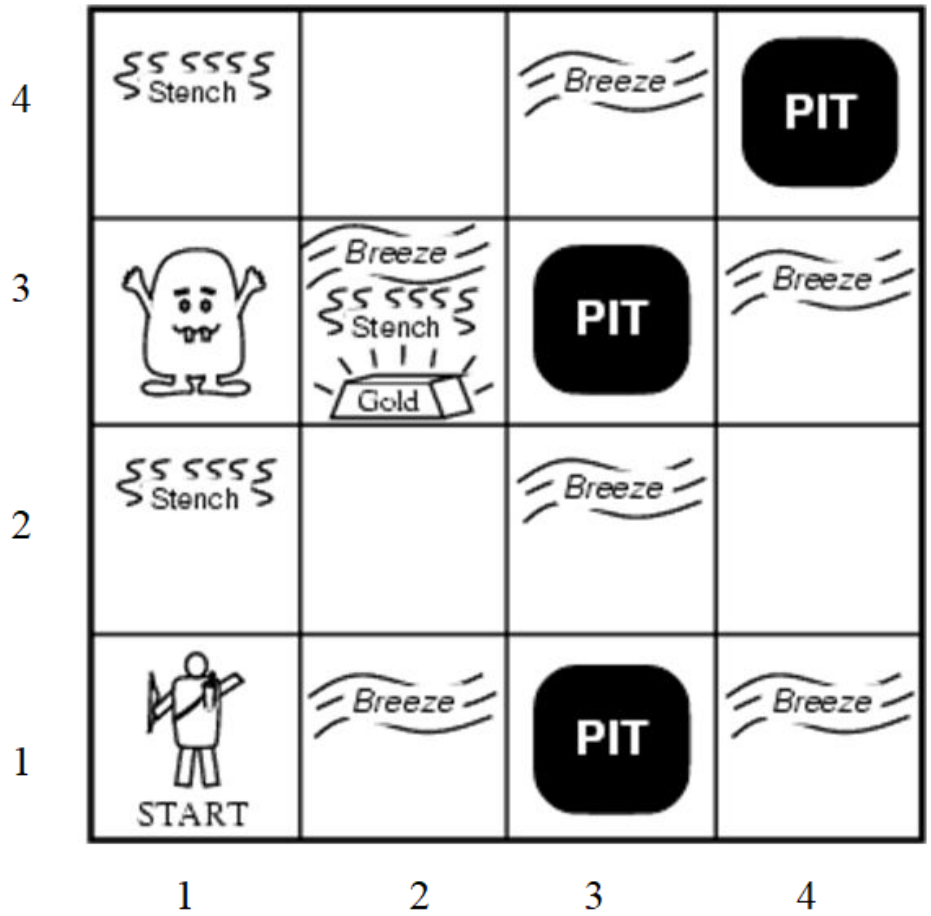
Propositional Logic

Recall: World Model

- World model: Encode information about
  - How the world evolves independently of the agent;
  - How the agent's actions affect the world.
- A structured representation is world model that describes the world by capturing:
  - Knowledge: Correlations between features.
  - Reasoning: Inferences that derive new knowledge from existing knowledge and percepts.



Example. [Wumpus world]



We need a structured representation that is able to code knowledge and percepts, and to perform inferences.

Recall: Knowledge-based Agent

Recall: Knowledge-based Agent

A knowledge-based agent is defined using a knowledge representation language and has the two main components:

- A knowledge base is a structured representation of the world specified by the knowledge representation language.
- A knowledge base can be updated through an inference engine.

KB Agent Program

Knowledge-based Agent Program Framework

INPUT: *percept*, background knowledge *KB*, clock t (initially 0)
OUTPUT: An action

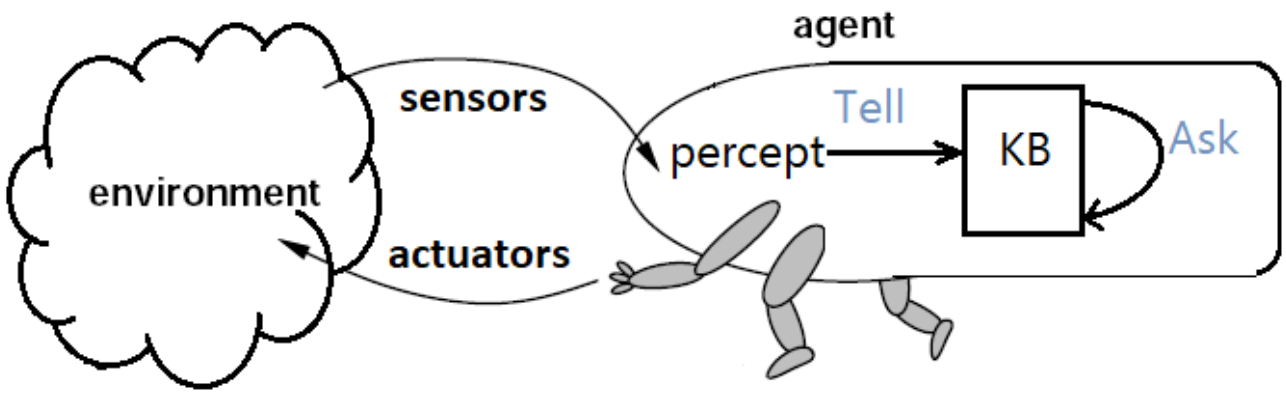
Tell(*KB*, MakePerceptSentence(*percept*, t))
action $\leftarrow$ Ask(*KB*, MakeActionQuery(*percept*, t))
return Tell(*KB*, MakeActionSentence(*action*, t)
$t \leftarrow t + 1$
return *action*

Two important operations:

- Tell: Adding a sentence to KB
- Ask: Querying if a sentence "follows" the KB

To design the two functions Tell and Ask, we need to answer the following questions:

1. What knowledge representation language to be used to define the KB?
2. How do we implement an IE using this language?



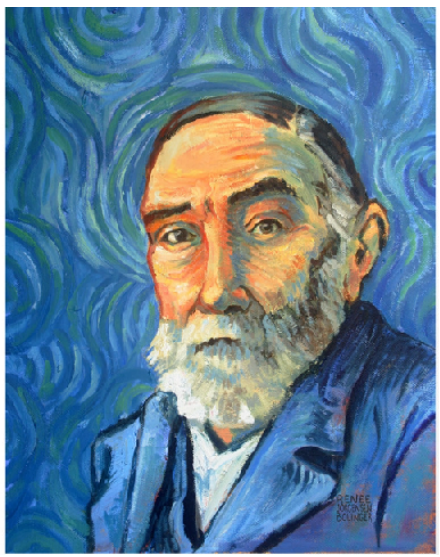
Question. How to define a knowledge representation language?

Observation. Knowledge is a constraint that is specified by a statement.

Example.

- 'There is breeze here' restricts the possible worlds to have breeze in the current location.
- 'At least one of these two locations must be a pit' restricts the possible worlds to have pit in either one or both of the two locations.
- 'If there is a stench, then a wumpus is nearby' restricts the possible worlds to link stench with wumpus.

- We need a language that expresses facts in the knowledge base as statements.
- The statements can be interpreted as constraints.
- An inference engine must be developed based on this framework that generates new statements based on logical rules.



Aristotle, 384BC–322BC, *Organon*. ⇝ George Boole, 1815–1864, *Mathematical Analysis of Logic*. ⇝ Gotlog Frege, 1848–1925, *Begriffsschrift*.

Question. How to define a language?

- **Atomic level**:
  - Syntax: What are the most elementary entities (i.e., atoms) in this language?
  - Semantic: How to interpret these atomic entities?
- **Compound level**:
  - Syntax: How to construct statements in this language?
  - Semantic: What is the meaning of a statement in this language?

Propositional Logic

Propositional logic provides a simple formal language framework. In this framework, we may specify sentences recursively:

- Atomic propositions: binary features of the form X with domain $\{\text{true}, \text{false}\}$ (or $\{0, 1\}$).
- Proposition: formed by connecting simpler propositions using connectives such as $\neg, \vee, \wedge, \rightarrow$.

Example. We have the following atomic propositions

- A: "It is raining"
- B: "The sun is shining"
- C: "There is rainbow"

Then $((A \wedge B) \rightarrow C)$ represents 'If it is raining and the sun is shining, then there is a rainbow"

Propositional Logic: Formal Syntax and Semantics

Definition [Atomic Entities]

- **Syntax**: Atomic propositions (or atoms) Atom $= \{X_1, \ldots, X_k\}$
- **Semantics**: An interpretation is a function $\pi\colon$ Atom $\to \{\text{true}, \text{false}\}$, i.e., π is a possible world.

Propositional Logic: Formal Syntax and Semantics

Definition [Atomic Entities]

- **Syntax**: Atomic propositions (or atoms) Atom $= \{X_1, \ldots, X_k\}$
- **Semantics**: An interpretation is a function $\pi\colon$ Atom $\to \{\text{true}, \text{false}\}$, i.e., π is a possible world.

Definition [Compound Entities]

- **Syntax**: A proposition can be made up from the atomic propositions in the form

$$\neg A, (A \vee B), (A \wedge B), (A \to B), (A \leftarrow B), (A \leftrightarrow B)$$

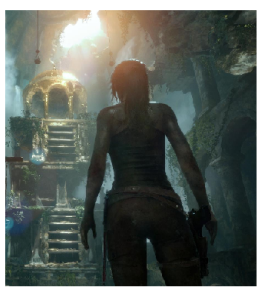
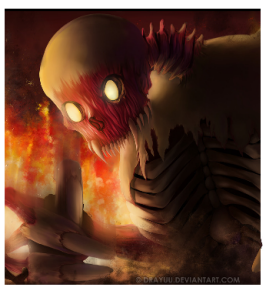
  where A, B are propositions, $\neg, \wedge, \vee, \to, \leftarrow, \leftrightarrow$ are connectives.
- **Semantics**: An interpretation determines the truth value of all propositions:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \leftarrow B$	$A \to B$	$A \leftrightarrow B$
true	true	false	true	true	true	true	true
true	false	false	false	true	true	false	false
false	true	true	false	true	false	true	false
false	false	true	false	false	true	true	true

Example. [Wumpus world] We introduce atomic propositions: For any location $(i, j) \in \{1, 2, 3, 4\}^2$

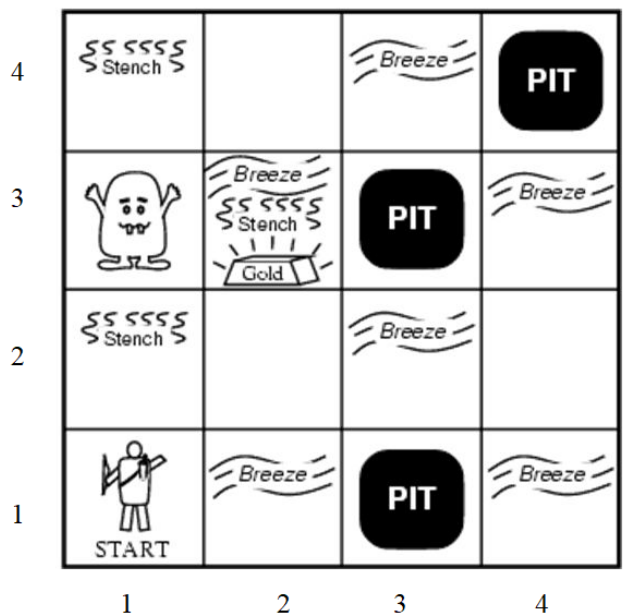
- $A_{i,j}$ is true if and only if the agent locates in square (i, j)
- $B_{i,j}$ is true if and only if square (i, j) has breeze
- $G_{i,j}$ is true if and only if square (i, j) contains gold
- $OK_{i,j}$ is true if and only if square (i, j) is safe
- $P_{i,j}$ is true if and only if square (i, j) has a pit
- $V_{i,j}$ is true if and only if square (i, j) has been visited by the agent
- $W_{i,j}$ is true if and only if square (i, j) has wumpus

These are the atomic propositions.

We can use formal, symbolic expressions for facts about the wumpus world:

- 'Square $(1,3)$ has a pit': $P_{1,3}$
- 'Square $(2,2)$ has no wumpus': $\neg W_{2,2}$
- 'Either $(2,2)$ has a pit or $(1,3)$ has a pit': $P_{2,2} \vee P_{1,3}$
- 'Since $(2,2)$ has no stench, $(1,2)$ has no wumpus': $\neg S_{2,2} \rightarrow \neg W_{1,2}$
- '$(2,4)$ is safe if and only if there is no pit nor wumpus on $(2,4)$': $OK_{2,4} \leftrightarrow (\neg P_{2,4} \wedge \neg W_{2,4})$



We can make inference from facts about the wumpus world:

- P_1 = 'There is no stench and no breeze in $(1,1)$': $\neg S_{1,1} \wedge \neg B_{1,1}$
- P_2 = 'There is no stench in $(1,1)$: $\neg S_{1,1}$

Note that from P_1, we can infer P_2.

- P_3 = 'If $(3,1)$ is a pit, then at least one of $(2,1)$ and $(3,2)$ has breeze'

  $P_{3,1} \rightarrow (B_{2,1} \vee B_{3,2})$
- P_4 = 'If none of $(2,1)$ and $(3,2)$ has no breeze, then $(3,1)$ is not a pit'

  $(\neg B_{2,1} \wedge \neg B_{3,2}) \rightarrow \neg P_{3,1}$

Note that from P_3, we can infer P_4, and vice versa.

Definition

A proposition A logically implies B, written $A \Rightarrow B$, if for any interpretation π,

$$\text{if } \pi(A) = \text{true then } \pi(B) = \text{true}.$$

Example.

- $\neg S_{1,1} \wedge \neg B_{1,1} \Rightarrow \neg S_{1,1}$
- $P_{3,1} \rightarrow (B_{2,1} \vee B_{3,2}) \Rightarrow (\neg B_{2,1} \wedge \neg B_{3,2}) \rightarrow \neg P_{3,1}$
- $(\neg B_{2,1} \wedge \neg B_{3,2}) \rightarrow \neg P_{3,1} \Rightarrow P_{3,1} \rightarrow (B_{2,1} \vee B_{3,2})$
- For any proposition A, we have $A \Rightarrow A$.
- (Modus ponens) For any A, B, $((A \rightarrow B) \wedge A) \Rightarrow B$.
- (Modus tollens) For any A, B, $((A \rightarrow B) \wedge \neg B) \Rightarrow \neg A$.
- (Syllogism) For any A, B, C, $((A \rightarrow B) \wedge (B \rightarrow C)) \Rightarrow (A \rightarrow C)$

Two propositions A, B are logically equivalent, written $A \Leftrightarrow B$, if we have both $A \Rightarrow B$ and $B \Rightarrow A$.

$\neg\neg A$	$\Leftrightarrow$	A	Double negation
$(A \wedge B)$	$\Leftrightarrow$	$(B \wedge A)$	Commutative
$(A \vee B)$	$\Leftrightarrow$	$(B \vee A)$	law
$(A \wedge (B \wedge C))$	$\Leftrightarrow$	$((A \wedge B) \wedge C)$	Associative
$(A \vee (B \vee C))$	$\Leftrightarrow$	$((A \vee B) \vee C)$	law
$(A \wedge (B \vee C))$	$\Leftrightarrow$	$((A \wedge B) \vee (A \wedge C))$	Distributive
$(A \vee (B \wedge C))$	$\Leftrightarrow$	$((A \vee B) \wedge (A \vee C))$	law
$(A \wedge A)$	$\Leftrightarrow$	A	Idempotent
$(A \vee A)$	$\Leftrightarrow$	A	law
$\neg(A \wedge B)$	$\Leftrightarrow$	$(\neg A \vee \neg B)$	De Morgan's
$\neg(A \vee B)$	$\Leftrightarrow$	$(\neg A \wedge \neg B)$	law
$(A \rightarrow B)$	$\Leftrightarrow$	$(\neg A \vee B)$	Implication
$(A \rightarrow B)$	$\Leftrightarrow$	$\neg(A \wedge \neg B)$	law
$(A \rightarrow B)$	$\Leftrightarrow$	$B \leftarrow A$	
$(A \vee (B \wedge \neg B))$	$\Leftrightarrow$	A	Contradiction
A	$\Leftrightarrow$	$(A \wedge (A \vee B))$	Absorption
A	$\Leftrightarrow$	$(A \vee (A \wedge B))$	law
$(A \leftrightarrow B)$	$\Leftrightarrow$	$((A \rightarrow B) \wedge (B \rightarrow A))$	Equivalence
$(A \leftrightarrow B)$	$\Leftrightarrow$	$((A \wedge B) \vee (\neg A \wedge \neg B))$	law

- An interpretation π satisfies a proposition A if $\pi(A) = \text{true}$.
- An interpretation π satisfies a set S of propositions if π satisfies every $A \in S$.

- An interpretation π satisfies a proposition A if $\pi(A) = \text{true}$.
- An interpretation π satisfies a set S of propositions if π satisfies every $A \in S$.

Definition

Suppose $X_1, \ldots, X_k$ are atomic propositions.

- A literal is X_i or $\neg X_i$.
- A clause is a proposition of the form

$$(h_1 \vee h_2 \vee \cdots \vee h_m) \leftarrow (\ell_1 \wedge \ell_2 \wedge \cdots \wedge \ell_k)$$

  where $m \geq 1$ and $k \geq 0$, each h_i and each ℓ_j is a literal.
- $(h_1 \vee h_2 \vee \cdots \vee h_m)$ is called the head of the clause.
  $(\ell_1 \wedge \ell_2 \wedge \cdots \wedge \ell_k)$ is called the body of the clause.

E.g. $\neg C \leftarrow (\neg A \wedge B)$ is a clause. $C \leftarrow (A \vee B)$ is not.

Note. A clause expresses a constraint:

The following propositions are logically equivalent

1. $(h_1 \vee h_2 \vee \cdots \vee h_m) \leftarrow (\ell_1 \wedge \ell_2 \wedge \cdots \wedge \ell_k)$
2. $h_1 \vee h_2 \vee \cdots \vee h_m \vee \neg\ell_1 \vee \neg\ell_2 \vee \cdots \vee \neg\ell_k$
3. $\neg(\neg h_1 \wedge \neg h_2 \wedge \cdots \wedge \neg h_m \wedge \ell_1 \wedge \ell_2 \wedge \cdots \wedge \ell_k)$

These propositions are equivalent to saying that

"$h_1, \ldots, h_m, \neg\ell_1, \ldots, \neg\ell_k$ should not be all false".

Therefore there is a close relationship between finding a satisfying interpretation of a proposition and constraint satisfaction problem (to be discussed next time).

Propositional Knowledge Base

Question. What is a knowledge base?

- A set of clauses can be regarded as knowledge about the atomic propositions.
- For any proposition A, there is a set of clauses S such that any interpretation π satisfies A if and only if π satisfies S.

Definition

A propositional knowledge base is a set of clauses.

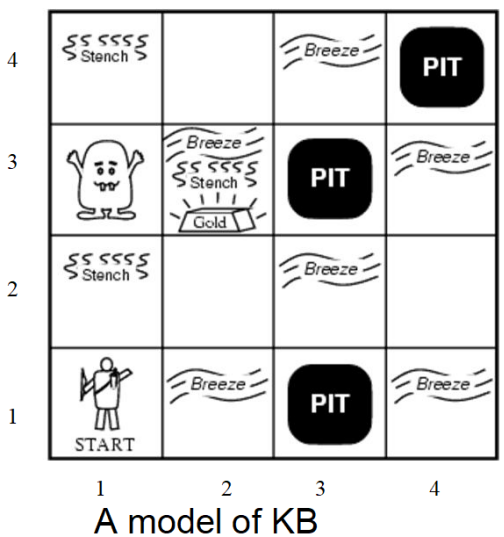
Definition

A model of a propositional knowledge base KB is an interpretation that satisfies KB.

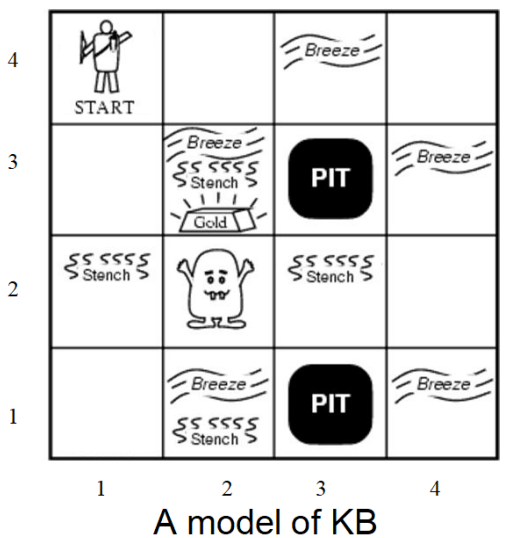
Example. [Wumpus world] KB contains

- $S_{i',j'} \leftarrow W_{i,j}$, and $B_{i',j'} \leftarrow P_{i,j}$ where $(i, j) \in \{1, \ldots, 4\}^2$ and $|i' - i| + |j' - j| = 1$.
- $(W_{i-1,j} \vee W_{i,j-1} \vee W_{i+1,j} \vee W_{i,j+1}) \leftarrow S_{i,j}$ and $(P_{i-1,j} \vee P_{i,j-1} \vee P_{i+1,j} \vee P_{i,j+1}) \leftarrow B_{i,j}$ for $i, j \in \{2, 3\}^2$.
- $(W_{2,1} \vee W_{1,2}) \leftarrow S_{1,1}$, $(P_{2,1} \vee P_{1,2}) \leftarrow B_{1,1}$, ... which describe if a location has wumpus or pit, then the surround locations have stench or breeze, respectively.

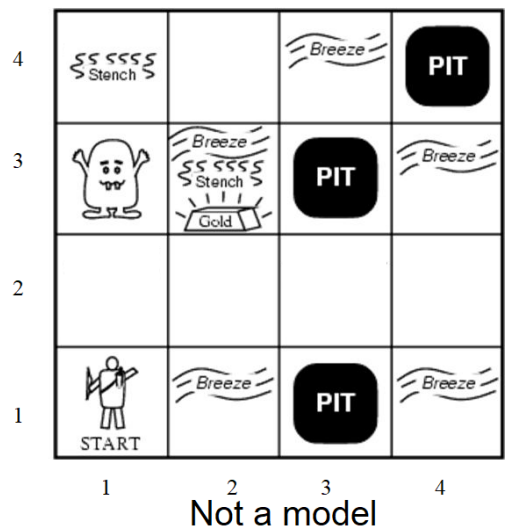
The following represents three interpretations:



A model of KB



A model of KB



Not a model

Propositional Inference Engine

Definition

A proposition g is called a logical consequence of a knowledge base KB, written as

$$\text{KB} \models g$$

if g is true in every model of KB.

Example. In the last example, $W_{1,2} \rightarrow S_{1,3}$ is a logical consequence of KB. So

$$\text{KB} \models S_{1,3} \leftarrow W_{1,2}$$

Similarly

$$\text{KB} \models \neg W_{2,2} \leftarrow \neg S_{3,2}$$

and

$$\text{KB} \models (W_{4,3} \vee W_{3,4}) \leftarrow (S_{3,3} \wedge \neg W_{2,3} \wedge \neg W_{3,2})$$

Question. What is an inference engine?

Definition

An inference engine decides for any KB, a set of percept atoms Percepts, and proposition g, whether

$$\text{KB} \cup \text{Percepts} \models g$$

The proposition g is called a query.

Summary of The Topic

The following are the main knowledge points covered:

- The framework of a **knowledge-based agent program**:
  - Tell: Adding a sentence to KB
  - Ask: Querying if a sentence is a logical consequence of the KB
- **Syntax** of propositional logic
  - Atomic propositions
  - (Composite) propositions
- **Semantics** of propositional logic
  - Interpretation
  - Truth table
- **Logical implications and equivalences**
- **Literals and clauses**
- **Model** of a propositional KB
- **Propositional logic inference engine:** $\text{KB} \cup \text{Percepts} \models g$