Artificial Intelligence & Machine Learning

A.Y. 2024/2025

Prof. Fabio Patrizi

Propositional Logic

What is AI?

AI is about devising *agents* (programs or devices) that act "Intelligently"

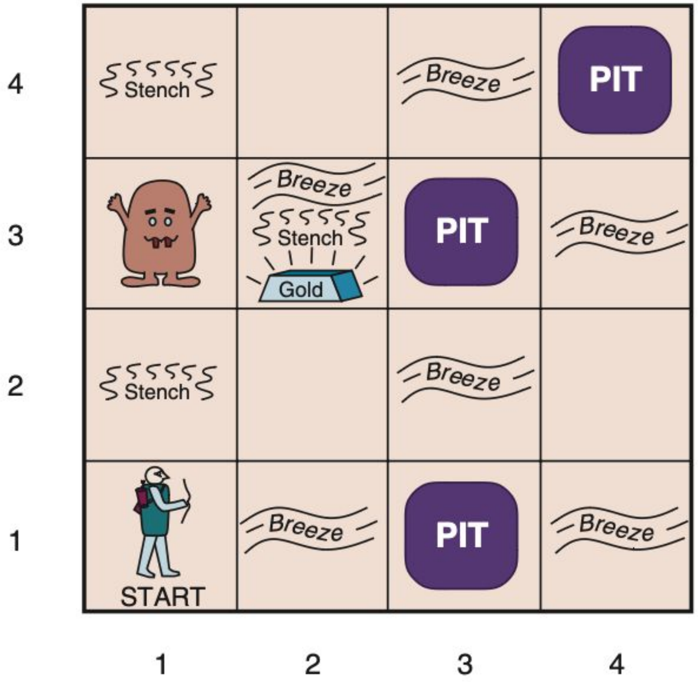
Blurry definition, informal

Several approaches, over time:

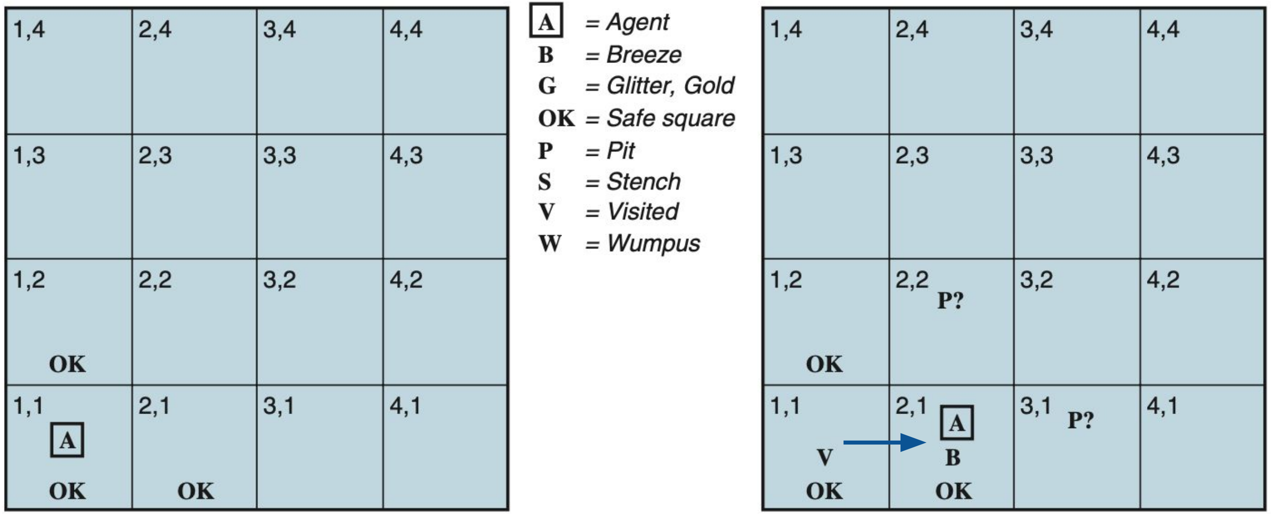
1. Turing test (1950): act as a human
   - Needs Knowledge Representation and Reasoning abilities (we have it!)
2. Cognitive modeling: think as a human
   - Do we really know how we think?
3. Think rationally: use rules of logic (e.g., Aristotle Syllogisms)
   - Needs formalization of logical reasoning (we have it!)
4. **Act rationally**: act so as to achieve best (expected) outcome
   - Prevailing approach to date, combines 1, 3 and additional abilities (e.g., autonomy, uncertainty)

Motivating Example: the Wumpus World

- **Environment**: A 4×4 grid of rooms, with walls surrounding the grid. The agent always starts in the square labeled [1,1], facing to the east. The locations of the gold and the wumpus are chosen randomly, with a uniform distribution, from the squares other than the start square. In addition, each square other than the start can be a pit, with probability 0.2.
- **Performance measure**: +1000 for climbing out of the cave with the gold, –1000 for falling into a pit or being eaten by the wumpus, –1 for each action taken, and –10 for using up the arrow. The game ends either when the agent dies or when the agent climbs out of the cave.
- **Actuators**: The agent can move *Forward*, *TurnLeft* by 90°, or *TurnRight* by 90°. The agent dies a miserable death if it enters a square containing a pit or a live wumpus. (It is safe, albeit smelly, to enter a square with a dead wumpus.) If an agent tries to move forward and bumps into a wall, then the agent does not move. The action *Grab* can be used to pick up the gold if it is in the same square as the agent. The action *Shoot* can be used to fire an arrow in a straight line in the direction the agent is facing. The arrow continues until it either hits (and hence kills) the wumpus or hits a wall. The agent has only one arrow, so only the first *Shoot* action has any effect. Finally, the action *Climb* can be used to climb out of the cave, but only from square [1,1].
- **Sensors**: The agent has five sensors, each of which gives a single bit of information:
  - In the squares directly (not diagonally) adjacent to the wumpus, the agent will perceive a *Stench*.[1]
  - In the squares 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.



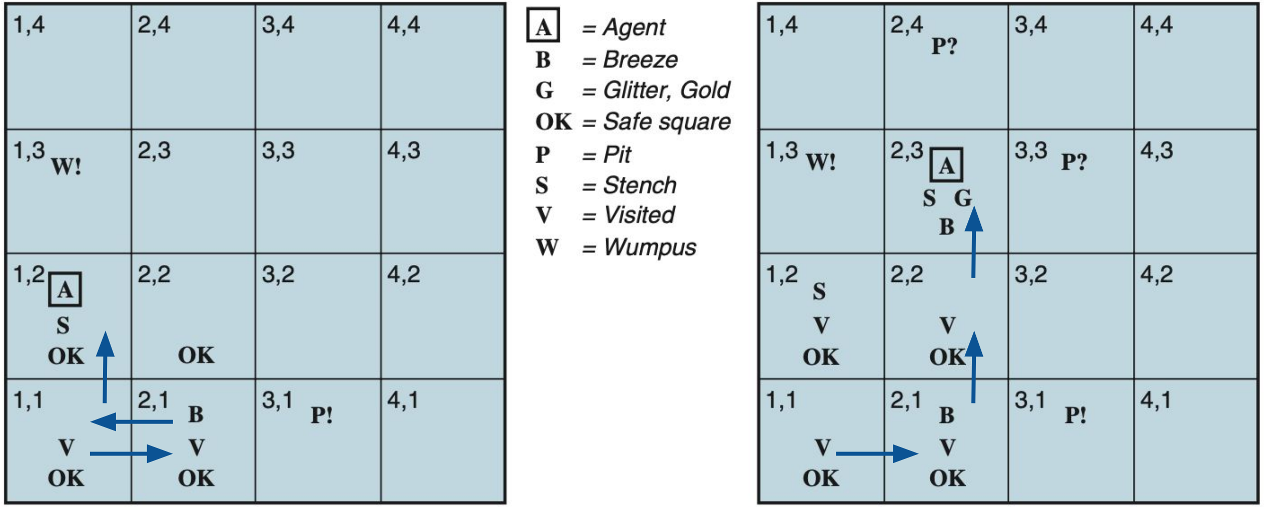
Wumpus World



initial state

[1,1] -> [2,1]

Wumpus World



[1,1] -> [2,1] -> [1,1] -> [1,2]

[1,1] -> [2,1] -> [2,2] -> [2,3]

Wumpus World

- Agent cannot see pits or Wumpus
- Can only perceive breeze and stench
- Yet, it is able to infer position of pits and Wumpus indirectly
- How can the agent do so?
- Can we systematize the inference procedure?
- Yes, with logic!

Propositional Logic

- Formalism for *representing* the state of the world and *reasoning* about it
- Based on *Propositions:*
  - Statements about the world which can be either *true* or *false*
  - E.g.:
    - Proposition $s_{1,2}$: "there is stench in cell [1,2]"
    - Proposition $p_{3,3}$: "there is a pit in cell [3,3]"
    - Proposition $b_{3,1}$: "there is breeze in cell [3,1]"
    - …

Building Blocks of Logical (Formal) Systems

Any Logical System must provide:

- Syntax: when does a string of characters is a *formula* in the logic?
- Semantics: what is the *meaning* of each formula?
- Inference system: how can we *infer* new knowledge?
    - E.g.:
        - We know that: stench in [1,2], no Wumpus in [1,1], no Wumpus in[2,2]
        - Then we can infer: Wumpus in [1,3]

Syntax of Propositional Logic

Propositional Alphabet:

- A (countable) set *P* of *atomic propositions* (or *atoms or variables*)
    - $s_{1,2}$, $p_{3,3}$, wumpus_in_1_3,...
    - This is typically defined by the user (and is usually, but not always, finite)
- Logical *connectives*:
    - $\neg$ negation (not)
    - $\land$ conjunction (and)
    - $\lor$ disjunction (or)
    - $\supset$ implication, also written as $\rightarrow$ (if then)
    - $\equiv$ (material) equivalence (if and and only if, iff), also written as $\leftrightarrow$
- Parenthesis symbols:
    - "(" and ")"

Syntax of Propositional Logic

Formulas of Propositional Logic are sequences of symbols from alphabet

Not all sequences are Propositional Logic formulas

The set of formulas is defined *inductively*

Formulas of Propositional Logic

- Every proposition $p \in P$ is a propositional formula (called *atomic formula*)
- If φ and ψ are propositional formulas then the following are propositional formulas:
  - $\neg\psi$
  - $\varphi \land \psi$
  - $\varphi \lor \psi$
  - $\varphi \supset \psi$
  - $\varphi \leftrightarrow \psi$
  - (φ)
- Nothing else is a propositional formula*

*can be omitted by saying that the set of formulas is define inductively

Formulas of Propositional Logic

These are (propositional) formulas:

- $s_{1,2}$ - *there is stench in [1,2]*
- $s_{1,2} \wedge p_{3,3}$ - *there is stench in [1,2] and a pit in [3,3]*
- $s_{1,2} \vee p_{3,3}$ - *there is stench in [1,2] or a pit in [3,3]*
- $b_{2,1} \supset p_{1,1} \vee p_{2,2} \vee p_{3,2}$ - *if breeze in [2,1] then pit in [1,1] or [2,2] or [3,2]*
-

These are not formulas (just sequences of propositional symbols):

- $\wedge p$
- $(qp \wedge \neg r \vee \wedge (pp$

Operator Precedence

How do we read formula $\varphi \land \psi \supset \gamma$?

- $(\varphi \land \psi) \supset \gamma$ (φ and ψ (together) imply γ)?
- $\varphi \land (\psi \supset \gamma)$ (φ and ψ implies γ)?

Solved using operator precedence:

1. $\neg$
2. $\land$
3. $\lor$
4. $\supset$
5. $\leftrightarrow$

Thus $\varphi \land \psi \supset \gamma$ is to be read as: $(\varphi \land \psi) \supset \gamma$, because $\land$ has higher precedence than $\supset$

Semantics of Propositional Logic

Semantics defines when a formula is true or false, depending on *whether the propositional symbols it contains are true or false*

In order to assign a truth value to a formula, we need to know the truth value of its symbols

E.g., whether formula $P \wedge Q$ is true depends on truth value of P and Q

Propositional Interpretations

A (propositional) *interpretation* is a function I assigning T (true) or F (false) to every propositional symbol, i.e., I: $P \rightarrow$ {T,F}

Example: $I(s_{1,2})$=; $I(b_{2,2})$=F; $I(b_{1,2})$=T

We represent an interpretation I as a set $I \subseteq P$ under the convention that:

$$I(p)=T \text{ if and only if } p \in I$$

Example:

I={$s_{1,2}$,$b_{1,2}$} represents interpretation I above

Propositional interpretations are sometimes called *propositional models*

Semantics of Propositional Logic

We define when a formula φ is true under an interpretation *I* (written $I \models \varphi$) inductively as follows:

- if φ is an atom, then $I \models \varphi$ iff $\varphi \in I$
- if $\varphi = \neg\psi$ then if $I \models \varphi$ iff ~~ψ is false~~ $I \not\models \psi$
- if $\varphi = \psi \wedge \gamma$ then $I \models \varphi$ iff $I \models \psi$ and $I \models \gamma$
- if $\varphi = \psi \vee \gamma$ then $I \models \varphi$ iff $I \models \psi$ or $I \models \gamma$
- if $\varphi = \psi \supset \gamma$ then $I \models \varphi$ iff $I \not\models \psi$ or $I \models \gamma$
- if $\varphi = \psi \leftrightarrow \gamma$ then $I \models \varphi$ iff $I \models \psi \supset \gamma$ and $I \models \gamma \supset \psi$
- if $\varphi = (\psi)$ then $I \models \varphi$ iff $I \models \psi$

When $I \models \varphi$ we also say that *I* *satisfies* φ and call *I* a *model* of φ

Truth Tables

The semantics of propositional formulas can also be defined using truth tables

$\neg\psi \vee \gamma$

ψ	γ	$\neg\psi$	$\psi \wedge \gamma$	$\psi \vee \gamma$	$\psi \supset \gamma$	$\psi \leftrightarrow \gamma$ *	(ψ)
F	F	T	F	F	T	T	F
F	T	T	F	T	T	F	F
T	F	F	F	T	F	F	T
T	T	F	T	T	T	T	T

* $\psi \supset \gamma$ AND $\gamma \supset \psi$

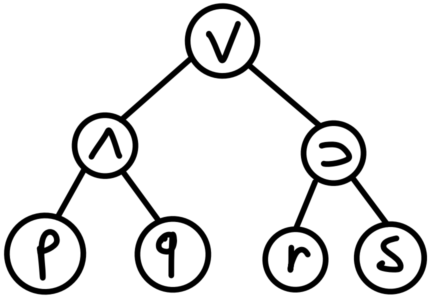
↳ $(\neg\psi \vee \gamma) \wedge (\neg\gamma \vee \psi)$

Semantics of Propositional Logic

For I={p}, does $I \models (p \wedge q) \vee (r \supset s)$? ✓

Let's apply the rules above:

- $I \models (p \wedge q) \vee (r \supset s)$ iff $I \models (p \wedge q)$ or $I \models (r \supset s)$
- $I \models (p \wedge q)$ iff $I \models p$ and $I \models q$
- $I \models p$ iff $p \in I$, thus $\boldsymbol{I \models p}$
- $I \models q$ iff $q \in I$, thus $\boldsymbol{I \not\models q}$
- since $I \not\models q$, we have that $\boldsymbol{I \not\models (p \wedge q)}$
- $I \models (r \supset s)$ iff $I \not\models r$ or $I \models s$
- $I \not\models r$ iff $r \notin I$, thus $\boldsymbol{I \not\models r}$ and hence $\boldsymbol{I \models (r \supset s)}$
- since $I \models (r \supset s)$ we have that $\boldsymbol{I \models (p \wedge q) \vee (r \supset s)}$



PROPOSITIONAL MODEL CHECK

Algorithm for Checking $I \vDash \varphi$

Boolean check(I, φ){

if φ is an atom then return true if $\varphi \in I$, *false otherwise*

if $\varphi = \neg\psi$ then return (!check(I, ψ))

if $\varphi = \psi \wedge \gamma$ then return (check(I, ψ) && check(I, γ))

if $\varphi = \psi \vee \gamma$ then return (check(I, ψ) || check(I, γ))

if $\varphi = \psi \supset \gamma$ then return ((not check(I, ψ)) || check(I, γ))

if $\varphi = \psi \leftrightarrow \gamma$ then return (check(I, $\psi \supset \gamma$) && check(I, $\gamma \supset \psi$))

if $\varphi = (\psi)$ then return check(I, ψ)

}

Useful Equivalences

The following semantic equivalences are easy to verify:

- $\psi \supset \gamma$ has the same meaning as $\neg\psi \vee \gamma$
- $\psi \supset \gamma$ has the same meaning as $\neg\gamma \supset \neg\psi$
- $\neg(\psi \vee \gamma)$ has the same meaning as $\neg\psi \wedge \neg\gamma$
- $\neg(\psi \wedge \gamma)$ has the same meaning as $\neg\psi \vee \neg\gamma$
- $\psi \supset \neg\gamma$ has the same meaning as $\neg\psi \vee \neg\gamma$

Wumpus World Knowledge Base

A set Γ of propositional formulas is also called a *Knowledge Base* (because it provides an agent with knowledge about some world)

The agent can use Γ to reason about the world

Let's build a simple Knowledge Base modeling the rules of the map

Wumpus World Knowledge Base

First, we need to define the atomic propositions

Atomic Propositions (omit gold for simplicity):

$p_{x,y}$ - pit in x,y

$w_{x,y}$- Wumpus in x,y

$b_{x,y}$ - breeze in x,y

$s_{x,y}$ - stench in x,y

$l_{x,y}$ - agent in in x,y

for $x,y \in \{1,2,3,4\}$

Wumpus World Knowledge Base

1. Wumpus is in **exactly** one square:

    $\bigvee w_{x,y}$ for x,y ∈ {1,2,3,4} (this says *in **at least one** square*)

    $\neg(w_{x,y} \wedge w_{x',y'})$ for (x,y)≠(x',y') (all these together say *in **at most** one square*)

Observe:

$\bigvee w_{x,y}$ for x,y ∈ {1,2,3,4} shortcut for:

$w_{1,1} \vee w_{1,2} \vee w_{1,3} \vee w_{1,4} \vee w_{2,1} \vee \ldots \vee w_{2,4} \vee w_{3,1} \vee \ldots \vee w_{3,4} \vee w_{4,1} \vee \ldots \vee w_{4,4}$ (same for $\bigwedge$)

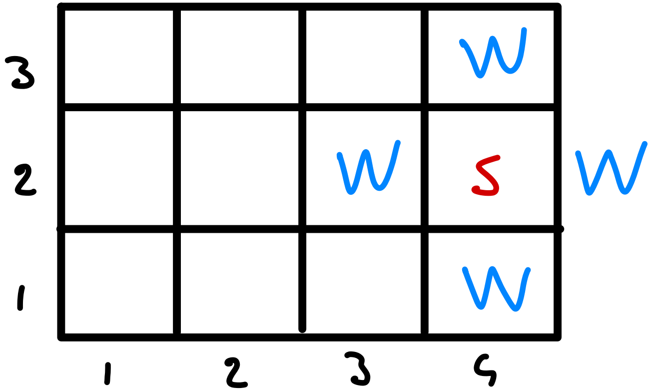
Wumpus World Knowledge Base

2. Square [x,y] has stench iff Wumpus in [x+1,y] or [x,y+1] or [x-1,y] or [x,y-1]

(for squares out of bounds proposition is dropped):

$$s_{x,y} \leftrightarrow w_{x+1,y} \vee w_{x,y+1} \vee w_{x-1,y} \vee w_{x,y-1}$$

$$S_{4,2} \leftrightarrow \cancel{W_{5,2}} \vee W_{4,3} \vee W_{3,2} \vee W_{4,1}$$



Wumpus World Knowledge Base

3. Square (x,y) has breeze iff pit in [x+1,y] or [x,y+1] or [x-1,y] or [x,y-1]

(for squares out of bounds proposition is dropped):

$$b_{x,y} \leftrightarrow p_{x+1,y} \vee p_{x,y+1} \vee p_{x-1,y} \vee p_{x,y-1}$$

4. No pit in [1,1]: $\neg p_{1,1}$

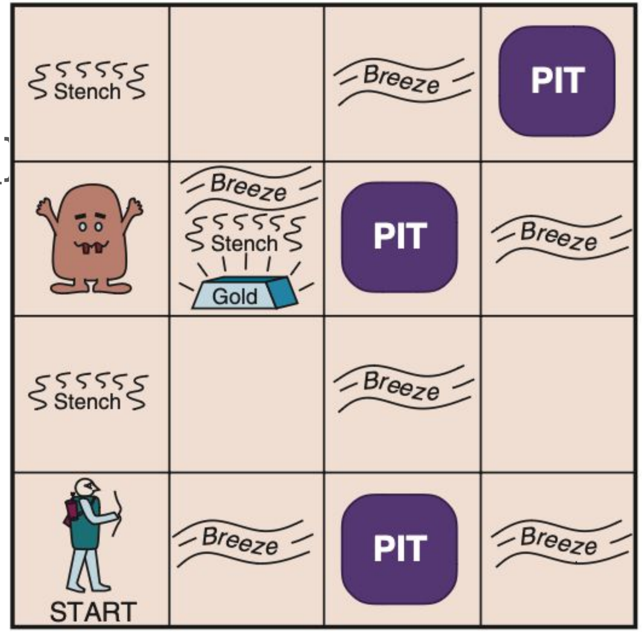
Logical Inference in Wumpus World

Every propositional interpretation corresponds to a map configuration. E.g., for map in figure:

$\{l_{1,1}, b_{2,1}, p_{3,1}, b_{4,1}, s_{1,2}, b_{3,2}, b_{2,3}, s_{2,3}, p_{3,3}, b_{4,3}, s_{1,4}, b_{3,4}, p_{4,4}$

But interpretations models also maps inconsistent with KB

Only the interpretations that satisfy **all** formulas in KB are relevant!

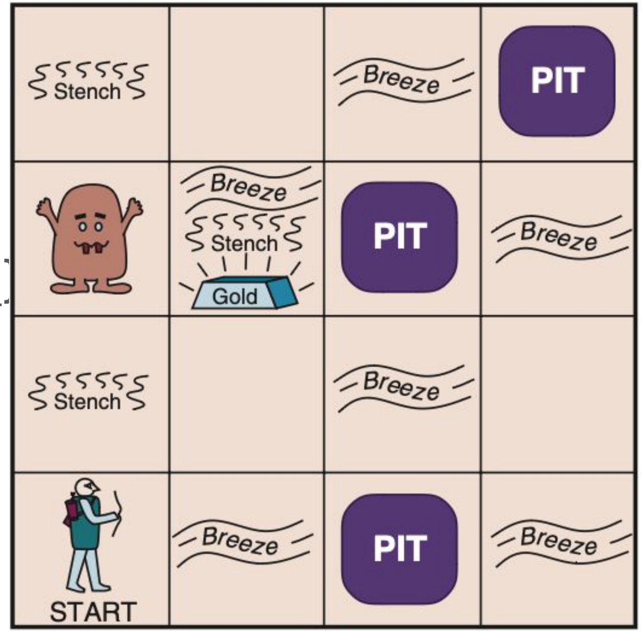


Logical Inference in Wumpus World

The interpretation associated with this map

satisfies all formulas in KB

$\{l_{1,1}, b_{2,1}, p_{3,1}, b_{4,1}, s_{1,2}, b_{3,2}, b_{2,3}, s_{2,3}, p_{3,3}, b_{4,3}, s_{1,4}, b_{3,4}, p_{4,4}$

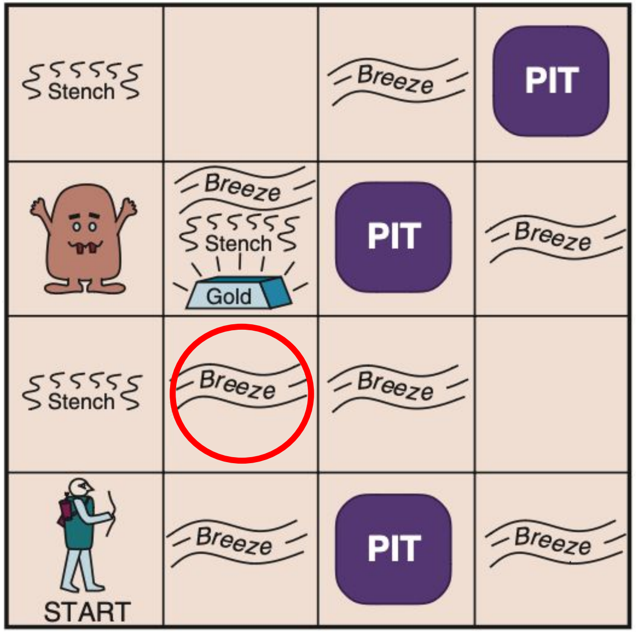


Logical Inference in Wumpus World

The interpretation associated with this map

does not satisfy all formulas in KB

$\{l_{1,1}, b_{2,1}, p_{3,1}, b_{4,1}, s_{1,2}, b_{3,2}, \mathbf{b_{2,2}}, b_{2,3}, s_{2,3}, p_{3,3}, b_{4,3}, s_{1,4}, b_{3,4}, p_{4,}$



The following formula is not satisfied:

$$b_{2,2} \leftrightarrow p_{3,2} \vee p_{2,3} \vee p_{1,2} \vee p_{2,1}$$

$\Gamma = \{\phi_1 \dots \phi_n\}$

CONSTRAINTS

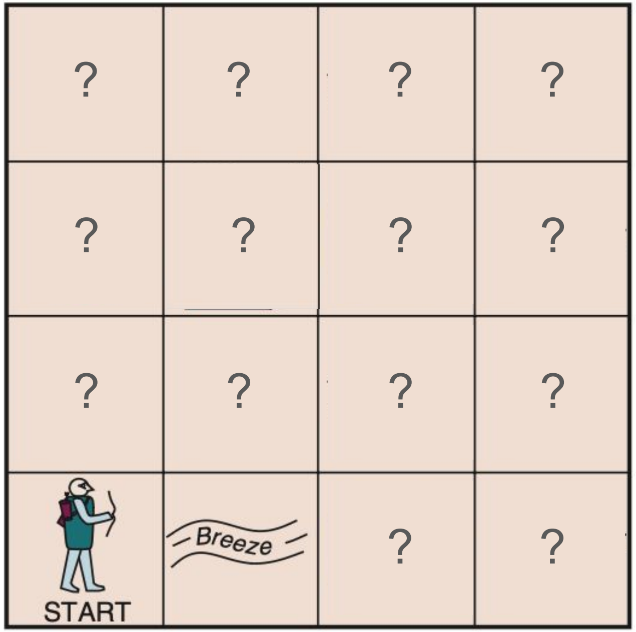
Logical Inference in Wumpus World

As the agent perceives features,some maps are ruled out, while other remain possible

This map cannot have a pit in [1,2], as the formula

$$b_{1,1} \leftrightarrow p_{1,2} \vee p_{2,1}$$

would not be satisfied, because the agent knows $\neg b_{1,1}$ thus both $p_{1,2}$ and $p_{2,1}$ must be false



$$I \models \ell_{11} \wedge b_{2,1} \wedge \phi_1 \wedge \dots \wedge \phi_n$$

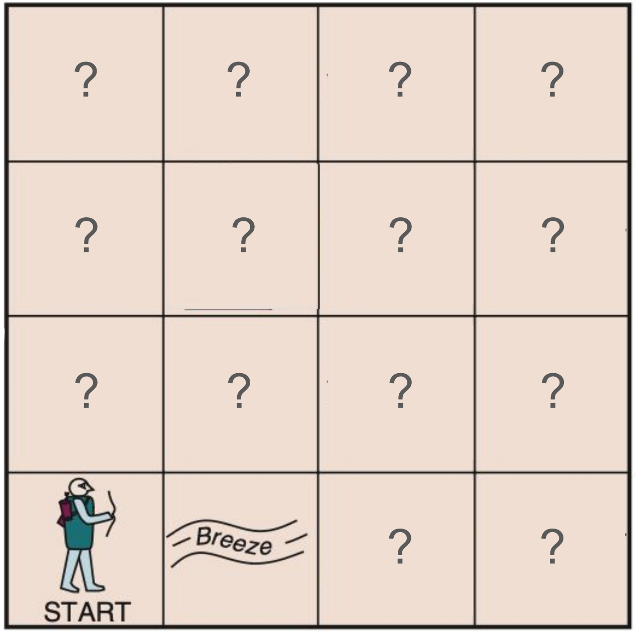
Logical Inference in Wumpus World

Before moving, agent checks if a pit can be in [1,2]

How?

- Consider observations collected so far:
    - $\{\neg b_{1,1}, \neg p_{1,1}, \neg w_{1,1}, b_{1,2}, \neg s_{1,1}, \neg s_{1,2}\}$
- Check whether there exists a model of **all** formulas in $\Gamma \cup \{\neg b_{1,1}, \neg p_{1,1}, \neg w_{1,1}, b_{1,2}, \neg s_{1,1}, \neg s_{1,2}\} \cup$ $\{P_{1,2}\}$

If one such model exists, there exists a valid map with breeze in [2,1], agent in [1,1] and pit in [1,2] (thus moving to [1,2] is unsafe)



Propositional Satisfiability

The problem of checking whether there exists an interpretation *I* that satisfies a propositional formula φ is called

Propositional Satisfiability Problem (SAT)

This is among the most (if not **the** most) studied problems in AI

It allows for checking whether a world *I* that satisfies certain properties is possible

Observation: checking whether $I \vDash \varphi_1$ and $I \vDash \varphi_2$,..., and $I \vDash \varphi_n$ is equivalent to

checking whether $I \vDash \varphi_1 \land \varphi_2 \land \ldots \land \varphi_n$ (applies only to finitely many formulas!)

Satisfiable Formulas

A propositional formula φ is said to be *satisfiable* if there exists an interpretation I such that $I \vDash \varphi$

If φ doesn't have any model, it is said to be *unsatisfiable*

Examples:

- formula $\neg(a \lor b) \to c$ is satisfiable: e.g., $I=\{c\}$
- formula $a \land (a \to b) \land (b \to \neg a)$ is unsatisfiable

Checking Satisfiability with Truth Tables

One naïve way to check a formula for satisfiability consists in constructing the truth table of the formula

The truth table lists all the interpretations of the propositions occurring in the formula and associates them with the truth value of the formula

To check satisfiability, we can check whether some interpretation exists which satisfies the formula

Checking Satisfiability with Truth Tables

Example: for $\neg(a \lor b) \rightarrow c$

All interpretations but the first one are models

x $\qquad$ $\neg x \lor c$

a	b	c	$(a \lor b)$	$\neg(a \lor b)$	$\neg(a \lor b) \rightarrow c$
F	F	F	F	T	F
F	F	T	F	T	T
F	T	F	T	F	T
F	T	T	T	F	T
T	F	F	T	F	T
T	F	T	T	F	T
T	T	F	T	F	T
T	T	T	T	F	T

$\exists I$ s.t. $I \models \varphi$?

YES, THE 2nd ROW

WE CAN HAVE 2^n ROW, WITH n VARIABLES

Checking Satisfiability with Truth Tables

Example: for $a \wedge (a \rightarrow b) \wedge (b \rightarrow \neg a)$

No interpretation is a model

a	b	¬a	(a→b)	(b→¬a)	a∧(a→b)∧(b→¬a)
F	F	T	T	T	F
F	T	T	T	T	F
T	F	F	F	T	F
T	T	F	T	F	F

Checking Satisfiability with Truth Tables

Checking satisfiability with truth tables is not the smartest approach

Requires listing all $2^{|P|}$ interpretations (P: propositions occurring in formula)

We shall see better approaches

Valid Formulas

A propositional formula φ is said to be *valid* if *I* ⊨ φ for all interpretations *I*

Examples:

- (a→b)∨(b→¬a) is valid
- (a→b) is satisfiable but not valid

φ IS SATISFIABLE IF $\exists I$ SUCH THAT $I \models \varphi$

φ IS VALID IF $\forall I$ SUCH THAT $I \models \varphi$

Checking Validity with Truth Tables

To check validity, we can check whether all interpretations satisfy the formula

Also for this, better approaches exist

Example: $(a \rightarrow b) \lor (b \rightarrow \neg a)$

a	**b**	**$(a \rightarrow b)$**	**$(b \rightarrow \neg a)$**	**$(a \rightarrow b) \lor (b \rightarrow \neg a)$**
F	F	T	T	T
F	T	T	T	T
T	F	F	T	T
T	T	T	F	T

Checking Validity with Truth Tables

Example: $(a \rightarrow b)$

a	**b**	**$(a \rightarrow b)$**
F	F	T
F	T	T
T	F	F
T	T	T

Satisfiability, Unsatisfiability, Validity

By the definitions of satisfiability, unsatisfiability and validity, we have that:

- if φ is valid then it is satisfiable
- if φ is unsatisfiable then it is not valid
- φ is satisfiable iff ¬φ is not valid
- φ is valid iff ¬φ is unsatisfiable
- φ is not valid iff ¬φ satisfiable
- φ is unsatisfiable iff ¬φ is valid

Proving the above is a simple exercise

Valid, Satisfiable and Unsatisfiable

$$A \to A$$
$$A \lor \neg A$$
$$\neg\neg A \leftrightarrow A$$
$$\neg(A \land \neg A)$$
$$A \land B \to A$$
$$A \to A \lor B$$

Valid

Satisfiable

$$A \lor B$$
$$A \to B$$
$$\neg(A \lor B) \to C$$
$$A \land \neg A$$
$$\neg(A \to A)$$
$$A \leftrightarrow \neg A$$
$$\neg(A \leftrightarrow A)$$

Not valid

Unsatisfiable

Satisfiable, Unsatisfiable and Valid Knowledge Bases

All previous notions can be lifted to KBs

An interpretation I is a *model* of (or *satisfies*) a KB $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$, written $I \vDash \Gamma$, if $I \vDash \varphi_1, \ldots, I \vDash \varphi_n$ (which is equivalent to $I \vDash \varphi_1 \land \ldots \land \varphi_n$ – only for finite KBs!)

A KB $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$ is:

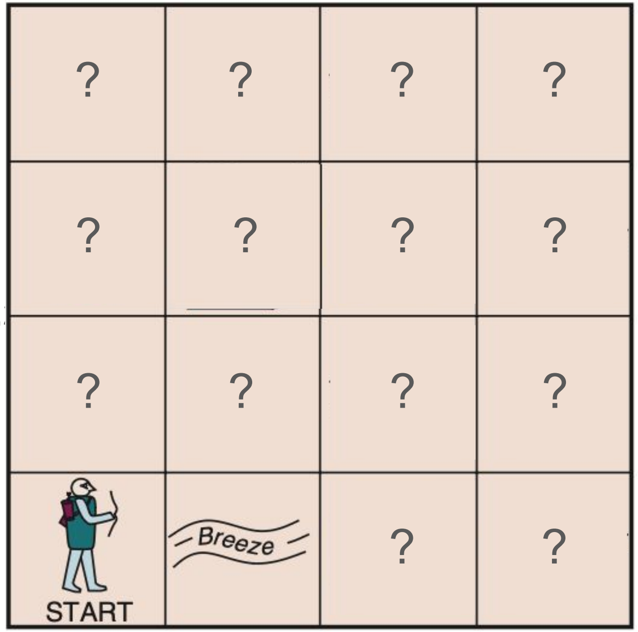
- *satisfiable* if there exists an interpretation I such that $I \vDash \Gamma$
- *unsatisfiable* if there exists no interpretation I such that $I \vDash \Gamma$
- *valid* if $I \vDash \Gamma$, for all interpretations I

Observation: all definitions apply also to infinite KBs

Using Satisfiability in Wumpus World

Let's check whether there *can* be a pit in [1,2]

- Assume observations collected so far:
    - $\{\neg b_{1,1}, \neg p_{1,1}, \neg w_{1,1}, b_{1,2}\}$
- Check whether there exists a model of **all** formulas in $\Gamma \cup \{\neg b_{1,1}, \neg p_{1,1}, \neg w_{1,1}, b_{1,2}\} \cup \{p_{1,}$



For simplicity, we consider only formulas from Γ mentioning squares [1,1], [1,2] or [2,1] (the only ones affected by observations)

Using Satisfiability in Wumpus World

Is this set of formulas satisfiable?

$w_{1,1} \lor w_{1,2} \lor w_{2,1} \lor w_{2,2}$

$\neg(w_{1,1} \land w_{1,2})$; $\neg(w_{1,1} \land w_{2,1})$; $\neg(w_{2,1} \land w_{1,2})$

$s_{1,1} \leftrightarrow w_{2,1} \lor w_{1,2}$; $s_{1,2} \leftrightarrow w_{1,1} \lor w_{2,2} \lor w_{1,3}$; $s_{2,1} \leftrightarrow w_{1,1} \lor w_{2,2} \lor w_{3,1}$

$b_{1,1} \leftrightarrow p_{2,1} \lor p_{1,2}$; $b_{1,2} \leftrightarrow p_{1,1} \lor p_{2,2} \lor p_{1,3}$; $b_{2,1} \leftrightarrow p_{1,1} \lor p_{2,2} \lor p_{3,1}$

Wumpus KB

$\neg b_{1,1}$; $\neg p_{1,1}$; $\neg w_{1,1}$; $b_{1,2}$; $\neg s_{1,1}$; $\neg s_{1,2}$ Collected Observations

$p_{1,2}$ Formula of interest

Using Satisfiability in Wumpus World

Assume *I* is an interpretation satisfying all the formulas above

Then, in particular:

1. $I \models p_{1,2}$
2. $I \models b_{1,1} \leftrightarrow p_{2,1} \vee p_{1,2}$

By 1., it follows that $p_{2,1} \in I$

By 2. and $p_{2,1} \in I$, it follows that $b_{1,1} \in I$

But if $b_{1,1} \in I$ then $I \not\models \neg b_{1,1}$

Therefore no *I* satisfying all formulas above may exist, thus **no pit can be in [1,2]**

Using Satisfiability in Wumpus World

We can also check whether Wumpus (instead of pit) can be in [1,2]:

$$\left.\begin{array}{l} w_{1,1} \lor w_{1,2} \lor w_{2,1} \lor w_{2,2} \\ \neg(w_{1,1} \land w_{1,2});\ \neg(w_{1,1} \land w_{2,1});\ \neg(w_{2,1} \land w_{1,2}) \\ s_{1,1} \leftrightarrow w_{2,1} \lor w_{1,2}\ ;\ s_{1,2} \leftrightarrow w_{1,1} \lor w_{2,2} \lor w_{1,3}\ ;\ s_{2,1} \leftrightarrow w_{1,1} \lor w_{2,2} \lor w_{3,1} \\ b_{1,1} \leftrightarrow p_{2,1} \lor p_{1,2}\ ;\ b_{1,2} \leftrightarrow p_{1,1} \lor p_{2,2} \lor p_{1,3}\ ;\ b_{2,1} \leftrightarrow p_{1,1} \lor p_{2,2} \lor p_{3,1} \end{array}\right\} \text{Wumpus KB}$$

$$\left.\neg b_{1,1}\ ;\ \neg p_{1,1}\ ;\ \neg w_{1,1};\ b_{1,2};\ \neg s_{1,1};\ \neg s_{1,2}\right\} \text{Collected Observations}$$

$$\left.w_{1,2}\right\} \text{Formula of interest}$$

Propositional Satisfiability

Whenever we want to check whether some (propositional) property is *possible*, we need to check for *satisfiability*

We want to automatize this procedure

Algorithms for satisfiability check (Later on)

Exercise

Check whether the following formulas are valid, satisfiable, or unsatisfiable

$$(p \to q) \land \neg q \to \neg p$$

$$(p \to q) \to (p \to \neg q)$$

$$(p \lor q \to r) \lor p \lor q$$

$$(p \lor q) \land (p \to r \land q) \land (q \to \neg r \land p)$$

$$(p \to (q \to r)) \to ((p \to q) \to (p \to r))$$

$$(p \lor q) \land (\neg q \land \neg p)$$

$$(\neg p \to q) \lor ((p \land \neg r) \leftrightarrow q)$$

$$(p \to q) \land (p \to \neg q)$$

$$(p \to (q \lor r)) \lor (r \to \neg p)$$

(Propositional) Logical Implication

We might also be interested in checking whether ***all models*** *that satisfy KB* Γ *and observations have a Wumpus in [1,3], i.e., whether we* ***are sure*** *Wumpus is there*

Formula φ *logically implies* ψ (φ⊨ψ), if for every *I* s.t. *I* ⊨ φ, it is the case that *I* ⊨ ψ

Corresponds to requiring that the set of models of ψ includes that of φ

(Notice we are *overloading* ⊨ for satisfaction and logical implication)

Can be lifted to KBs: Γ ⊨ ψ, if for every *I* s.t. *I* ⊨ Γ, it is the case that *I* ⊨ φ

Notice that ∅ ⊨ φ, simply written as ⊨ φ, is equivalent to saying that *φ is valid*

Logical Implication

Which ones of the following implications hold?

- $p \models p \vee q$
- $q \vee p \models p \vee q$
- $\{p \vee q, p \rightarrow r, q \rightarrow r\} \models r$
- $\{p \rightarrow q, p\} \models q$

1 $I \models p \Rightarrow I \models p \vee q$ ✓

2 IF $I \models p$ OR $I \models q \Rightarrow I \models p \vee q$ OR $I \models q \vee p$ ✓

3 WE MUST HAVE $p = F$ OR $q = F$ ✗

4

p	q	$\neg p \vee q$
0	0	1
0	1	1
1	0	0
1	1	1

$= q$ ✓

Logical Implication

Is there a Wumpus in [1,3]?

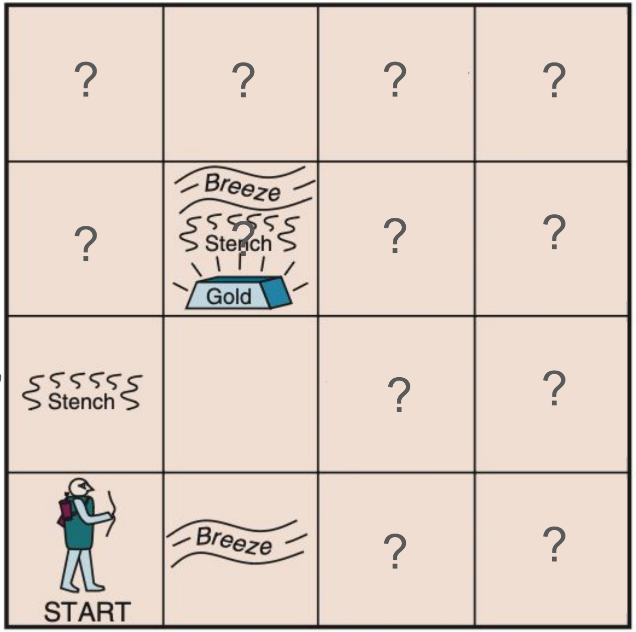
Wumpus-world KB: Γ

Observations:

O={¬$b_{1,1}$;¬$p_{1,1}$;¬$w_{1,1}$;¬$s_{1,1}$;$b_{2,1}$;¬$s_{2,1}$;¬$w_{2,1}$; …**¬$w_{2,2}$;$s_{1,2}$;$s_{2,}$**

Formula of interest: $w_{1,3}$ (Wumpus in [1,3])

Does $\Gamma \cup O \vDash w_{1,3}$?



Logical Equivalence

We can also be interested in checking whether two formulas (or KBs) have the *same meaning*.

φ and ψ are *logically equivalent* ($\varphi \equiv \psi$) if every interpretation *I* is s.t. $I \models \varphi$ iff $I \models \psi$

Corresponds to requiring that φ and ψ have *exactly the same models* and is equivalent to: $\varphi \models \psi$ *and* $\psi \models \varphi$

Logical Implication and Equivalence on KBs

Logical implication and logical equivalence can be lifted to KBs Γ and Γ'

$\Gamma \models \Gamma'$, if for every I s.t. $I \models \Gamma$, it is the case that $I \models \Gamma'$

(The set of models of Γ' includes that of Γ)

$\Gamma \equiv \Gamma'$, if $I \models \Gamma$ and $I \models \Gamma'$

(The set of models of Γ and Γ' match)

Properties of Propositional Logical Implication

Reflexivity: if $\varphi \in \Gamma$ then $\Gamma \vDash \varphi$

Ex falso (sequitur) quodlibet: if Γ is unsatisfiable then $\Gamma \vDash \varphi$ for every φ

Monotonicity: if $\Gamma \vDash \varphi$ then $\Gamma \cup \Gamma' \vDash \varphi$

Cut: if $\Gamma \vDash \varphi$ and $\Gamma' \cup \{\varphi\} \vDash \varphi'$ then $\Gamma \cup \Gamma' \vDash \varphi'$

Compactness: if $\Gamma \vDash \varphi$ then there is a finite $\Gamma' \subseteq \Gamma$ s.t. $\Gamma' \vDash \varphi$

Deduction Theorem: if $\Gamma \cup \{\varphi\} \vDash \varphi'$ then $\Gamma \vDash \varphi \to \varphi'$

Deduction Principle: $\Gamma \cup \{\varphi\} \vDash \varphi'$ iff $\Gamma \vDash \varphi \to \varphi'$

Refutation Principle: $\Gamma \vDash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable