KNOWLEDGE-BASED AGENTS & PROPOSITIONAL LOGIC

Lara J. Martin (she/they)
TA: Aydin Ayanzadeh (he)

9/28/2023
CMSC 671

By the end of class today, you will be able to:

1. Follow the "reasoning" of a simple knowledge-based agent
2. Use rules of logic to make inferences

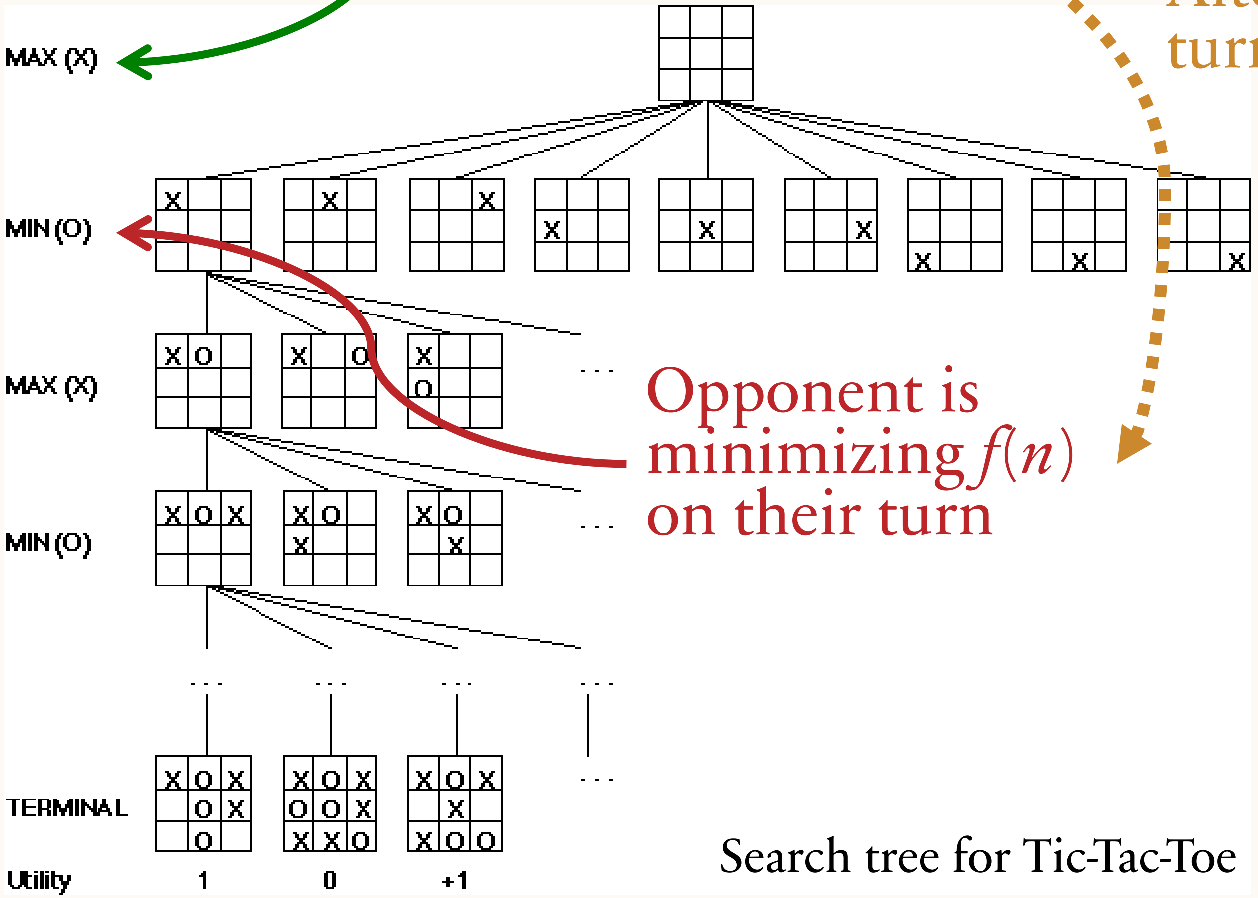
Modified from slides by Dr. Cynthia Matuszek & Dr. Chris Callison-Burch

RECAP

MINIMAX

I am maximizing $f(n)$ on my turn

Alternating turns

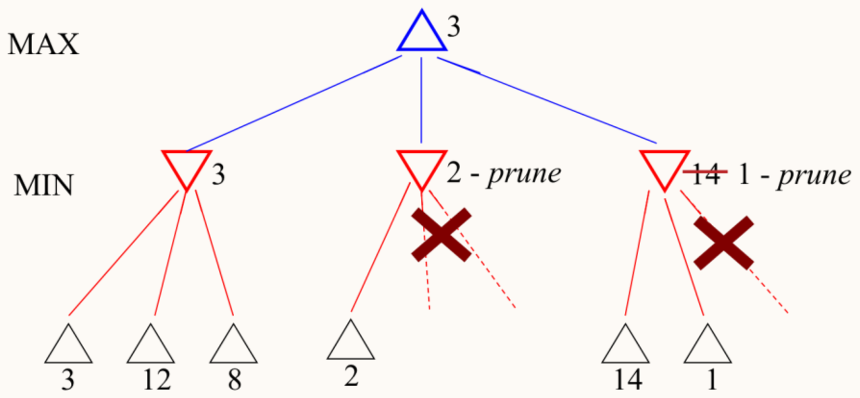


Opponent is minimizing $f(n)$ on their turn

Did we win, lose, or tie?

Search tree for Tic-Tac-Toe

RECAP



ALPHA-BETA PRUNING

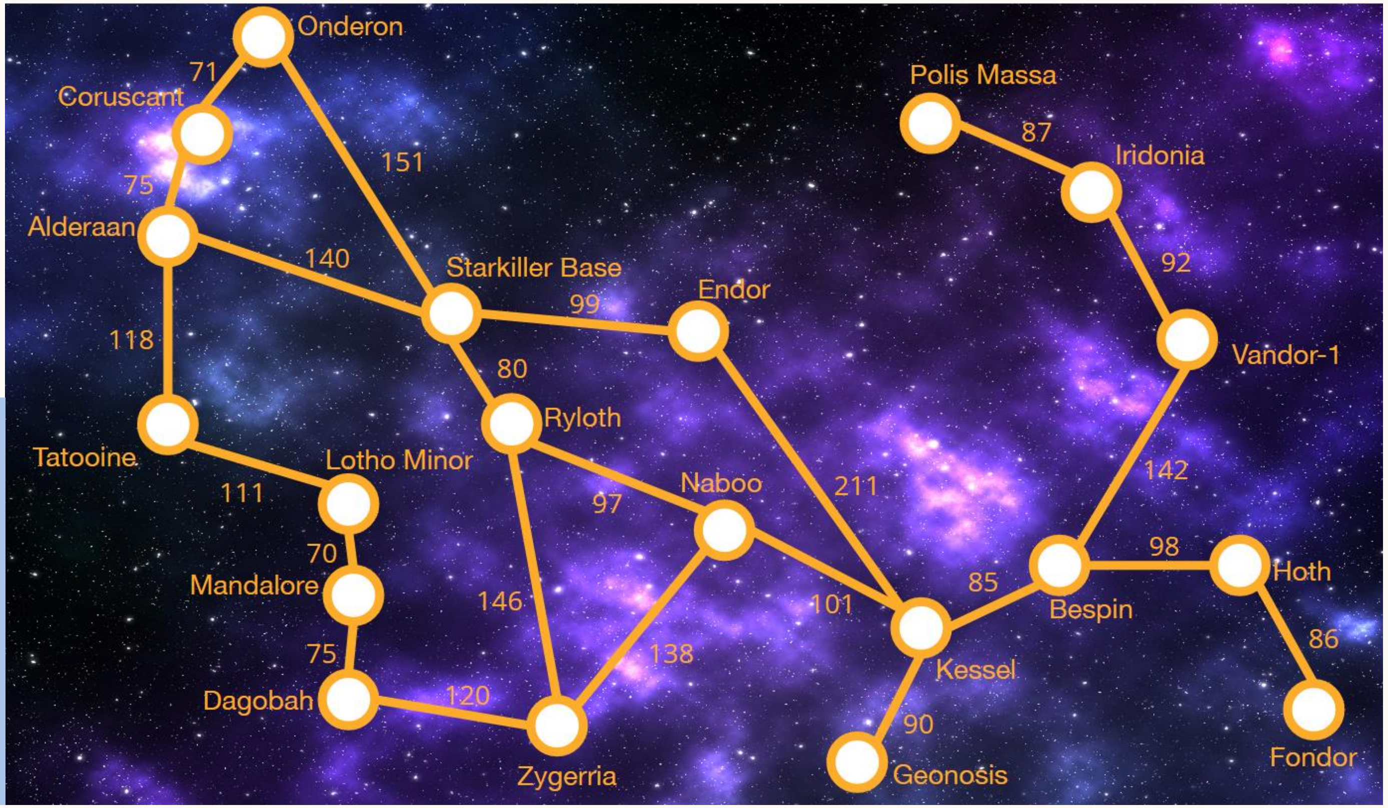
- At each **MAX** node n, α(n) = maximum value found so far
- At each **MIN** node n, β(n) = minimum value found so far
  - α starts at -∞ and increases, β starts at +∞ and decreases
- If **α > β, prune the branch**
  - i.e., stop searching that node's successors

KNOWLEDGE-BASED AGENTS

IN A GALAXY FAR, FAR AWAY…

So far, our problem-solving agents have performed a **search** over **states** to find a plan*. The representation of states has been **atomic**.

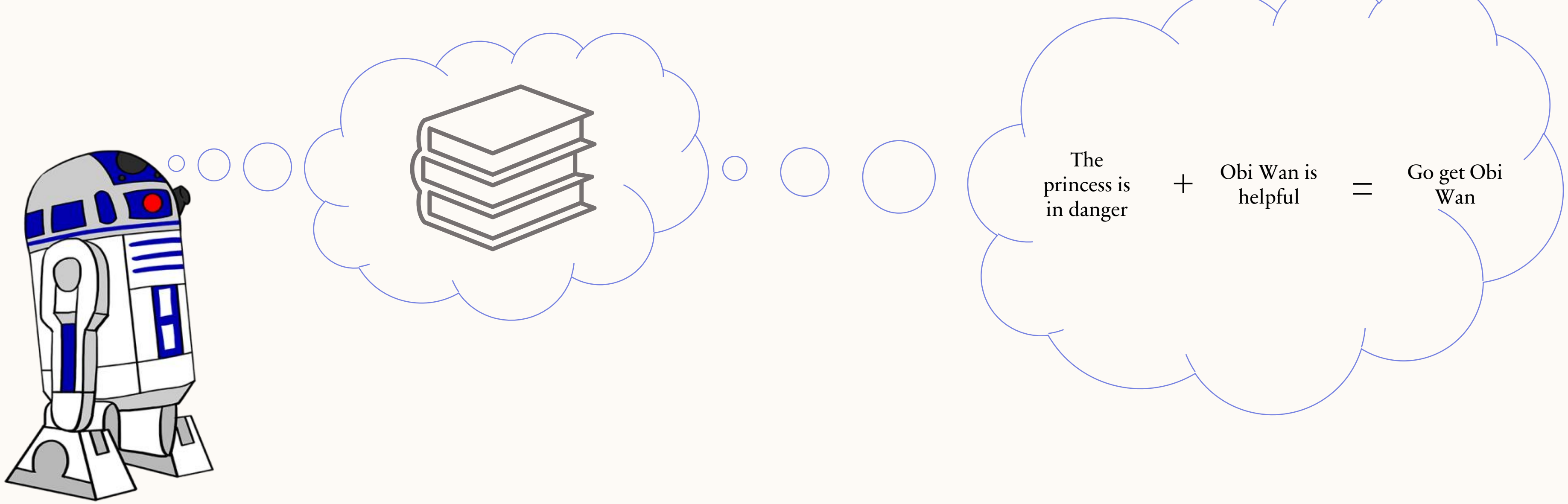
Limited to commands like
"Navigate to Kessel"
or
"Take me to the nearest habitable planet where I can store my perishable cargo"



*More on plans later in this module

WHAT IS A KNOWLEDGE-BASED AGENT?

- **Knowledge-based agents** use a process of **reasoning** over an explicit, internalized **representation** of knowledge to decide what action to take
- This set of knowledge is known as a **knowledge base**
- The agent can also **infer new things** from this knowledge



https://easydrawingguides.com/wp-content/uploads/2019/10/R2D2-10.png

KNOWLEDGE BASE (KB)

A KB contains a set of **sentences** (or **assertions**) that are written in a **knowledge representation language.** The sentence contains some assertion about the world.

Natural language sentences	Knowledge representation language sentence
Hoth is a planet	`planet(hoth)`
Hoth is habitable	`habitable(hot)`
Hoth is far from its sun	`far_from(hoth, sol)`
If a planet is far from its sun, then it is cold	`planet(x) and sun(y) and far_from(x, y) → cold (x)`

WHAT DOES A KB HOLD?

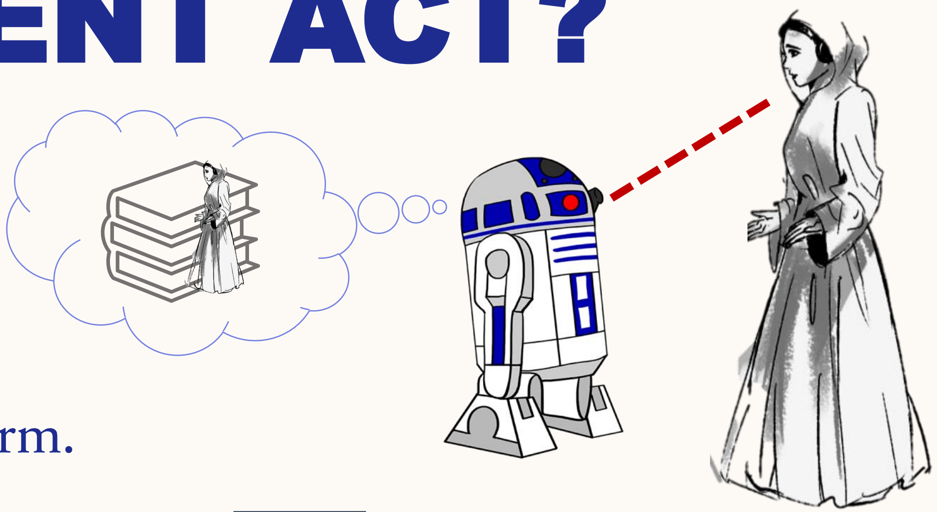
- The agent usually starts with some **background knowledge** about the world, then the agent can add to the information in the KB through its observations of the world.
- The agent can also query the KB and ask it to derive new knowledge in order to select what action it should take.
  - The process of deriving new sentences from old sentences is called **inference**.
- There are two kinds of sentences:
  - **Axioms** – a sentence that is given
  - **Derived sentences** – a new sentence that is derived from others sentences

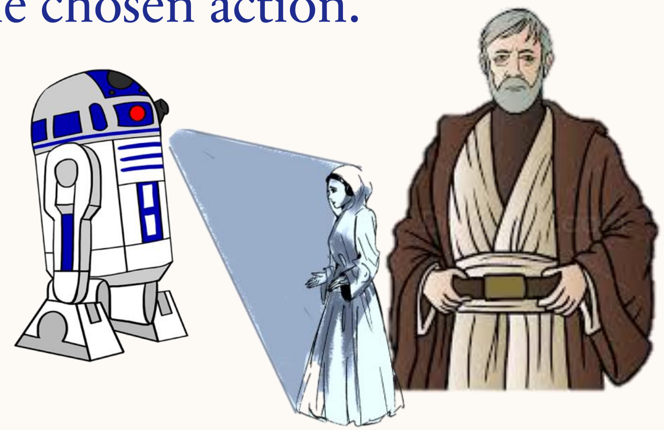
HOW DOES A KNOWLEDGE-BASED AGENT ACT?

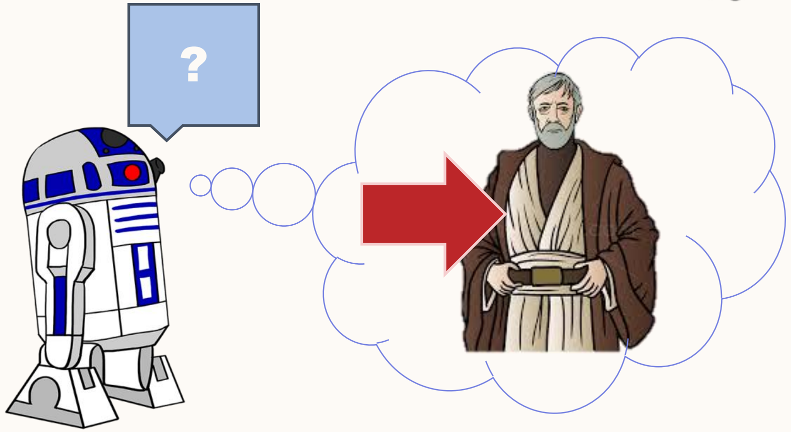
1. **TELLs** the knowledge base what it perceives.
   - (“asserts” knowledge into the KB)

2. **ASKs** the knowledge base what action to perform.
   - (performs “inference”)

3. **PERFORMs** the chosen action.







A SIMPLE KNOWLEDGE-BASED AGENT MUST…

- Represent states, actions, etc.
- Incorporate new percepts
- Update internal representations of the world
- Deduce hidden properties of the world
- Deduce appropriate actions

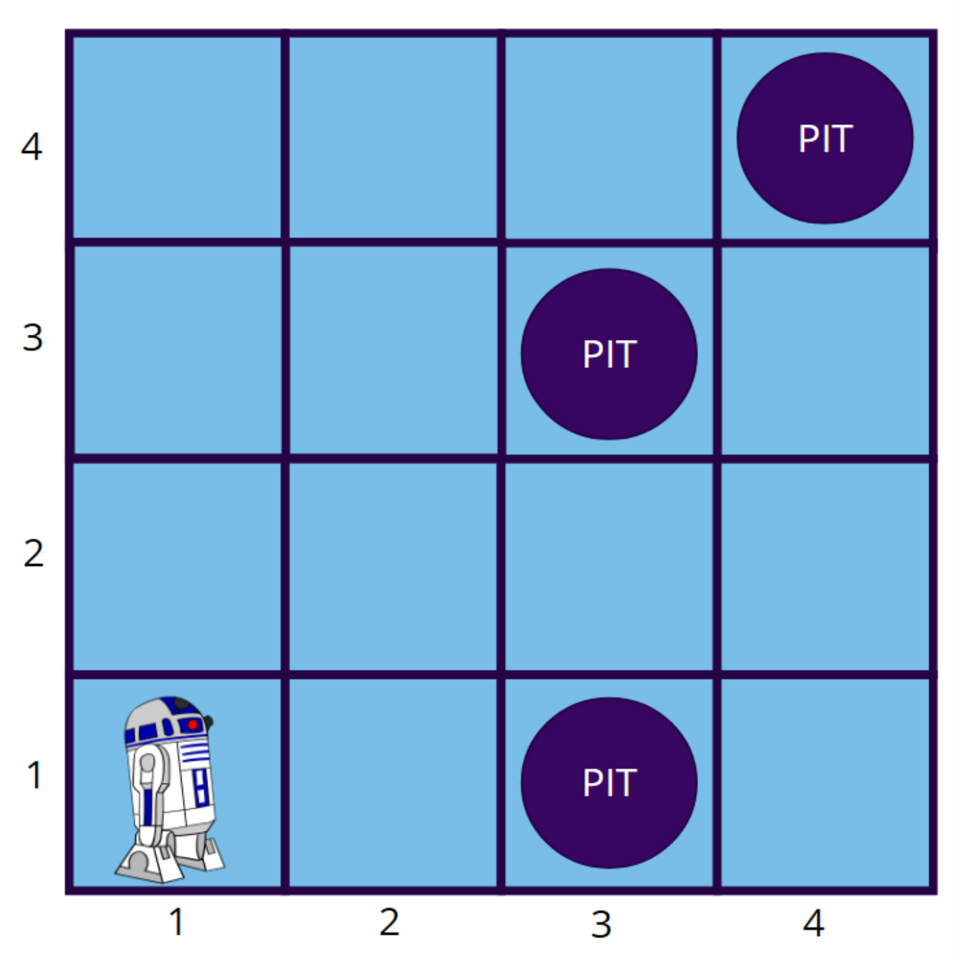
```
function KB-AGENT(percept) returns an action
  persistent: KB, a knowledge base
              t, a counter, initially 0, indicating time

  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t ← t + 1
  return action
```

EXAMPLE: WAMPA WORLD

- Our knowledge-based agent, R2D2, explores a cave consisting of rooms connected by passageways.
- Lurking somewhere in the cave is the Wampa, a beast that eats any agent that enters its room.
- Some rooms contain bottomless pits that trap any agent that wanders into the room.
- In one room is Luke
- The goal is:
  - collect Luke
  - exit the world
  - without being eaten

WAMPA WORLD ENVIRONMENT

Environment: A 4x4 grid of rooms. The agent starts in the square [1,1]. Wampa and Luke are randomly placed in other squares. Each square can be pit with 20% probability.



WAMPA WORLD PERFORMANCE MEASURE

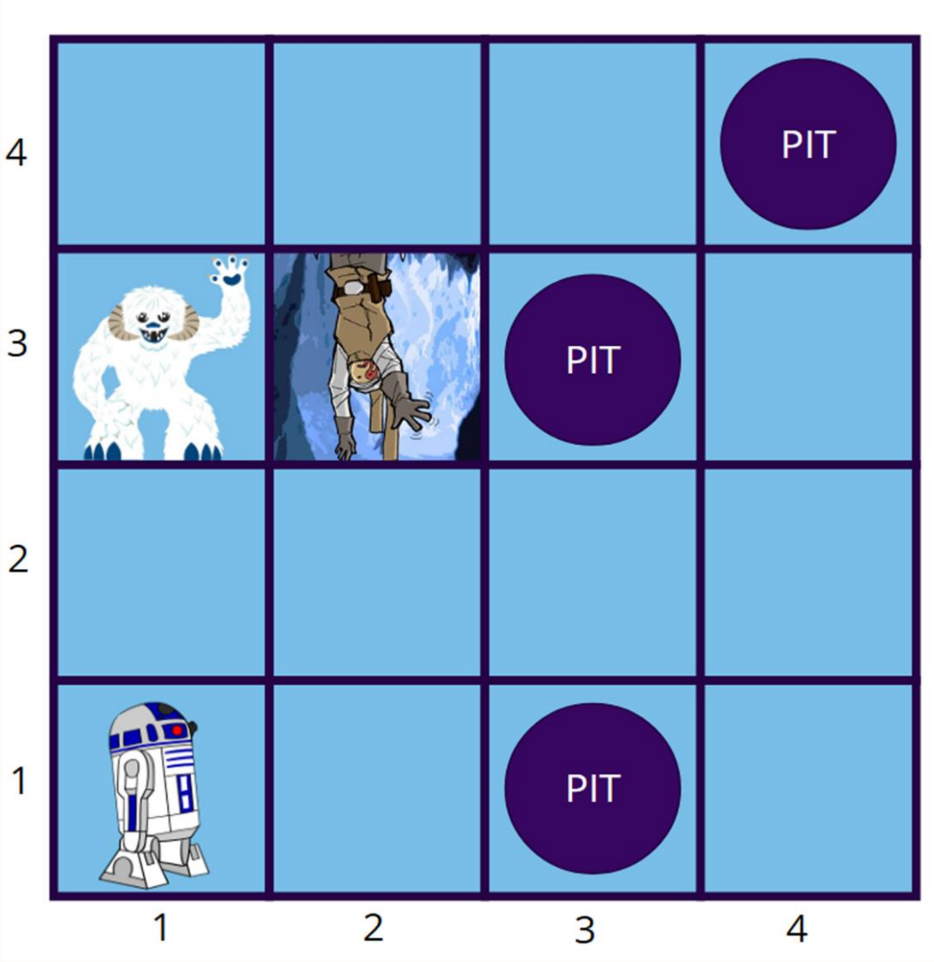
Performance measure:

+1000 points for rescuing Luke and leaving the cave

-1000 for falling into a pit or being eaten by the Wampa

-1 for each action taken

-10 for using up your blaster fire



WAMPA WORLD ACTUATORS

Actuators:

R2 can move *Forward*, *TurnLeft*, *TurnRight*.

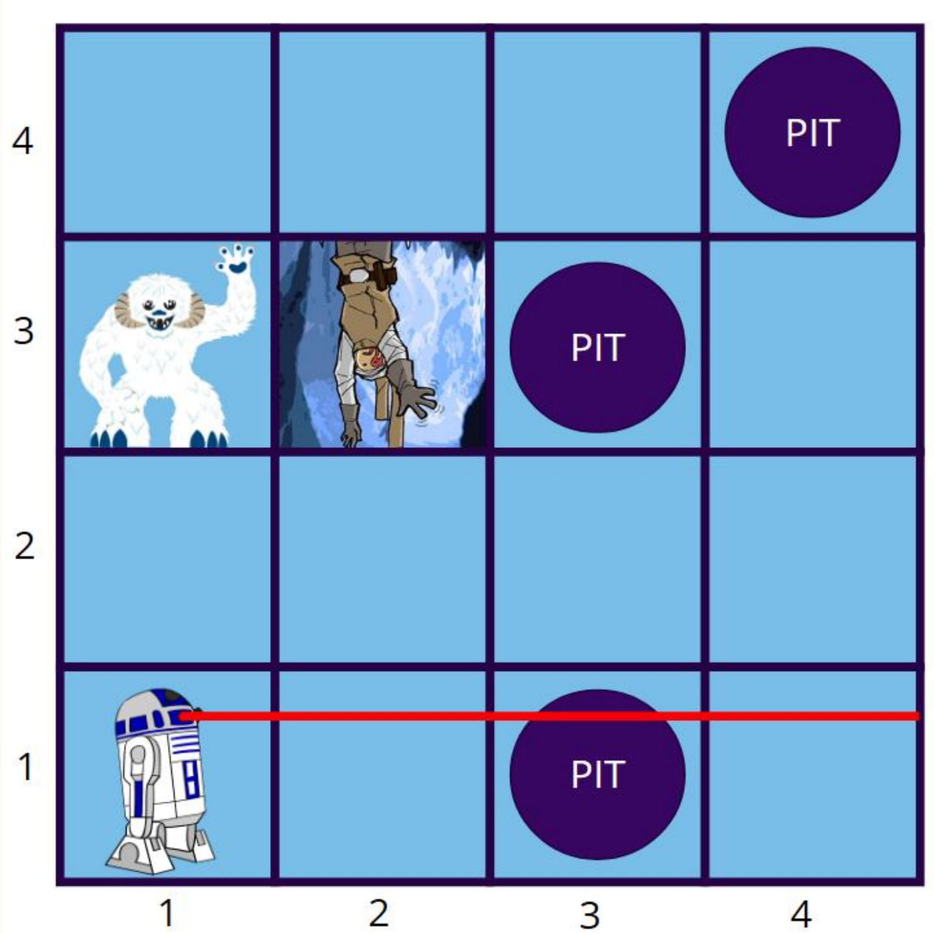
Agent dies if it moves into a pit or a Wumpus square.

Grab can pick up Luke.

Shoot fires blaster bolt in a straight line in the direction that R2D2 is facing.

If the blaster hits the Wampa, it dies. R2 only has enough power for one shot.

Climb gets R2 out of the cave but only works in [1, 1]



WAMPA WORLD SENSORS

Sensors:

In each square adjacent to the Wampa, R2D2's olfactory sensor perceives a *Stench*

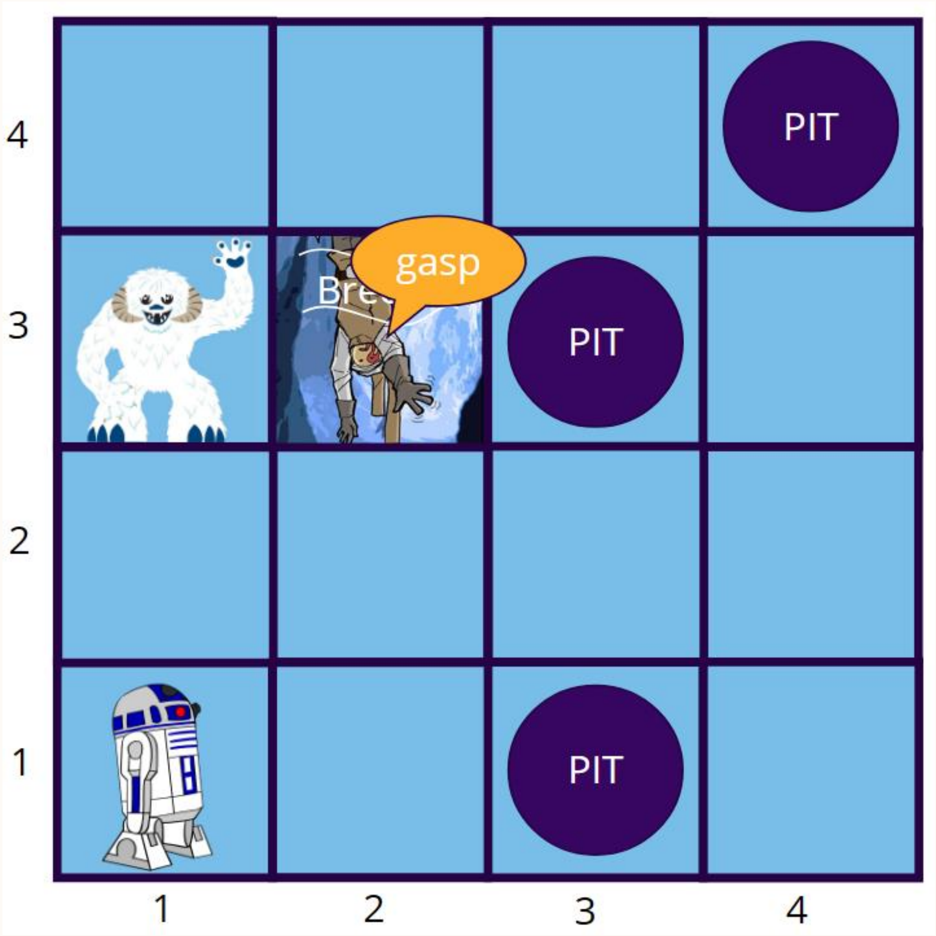
In each square adjacent to a pit, R2D2's wind sensor perceives a *Breeze*

In the square with Luke, R2D2's audio sensor perceives a *Gasp*

When R2D2 walks into a wall it perceives a *Bump*

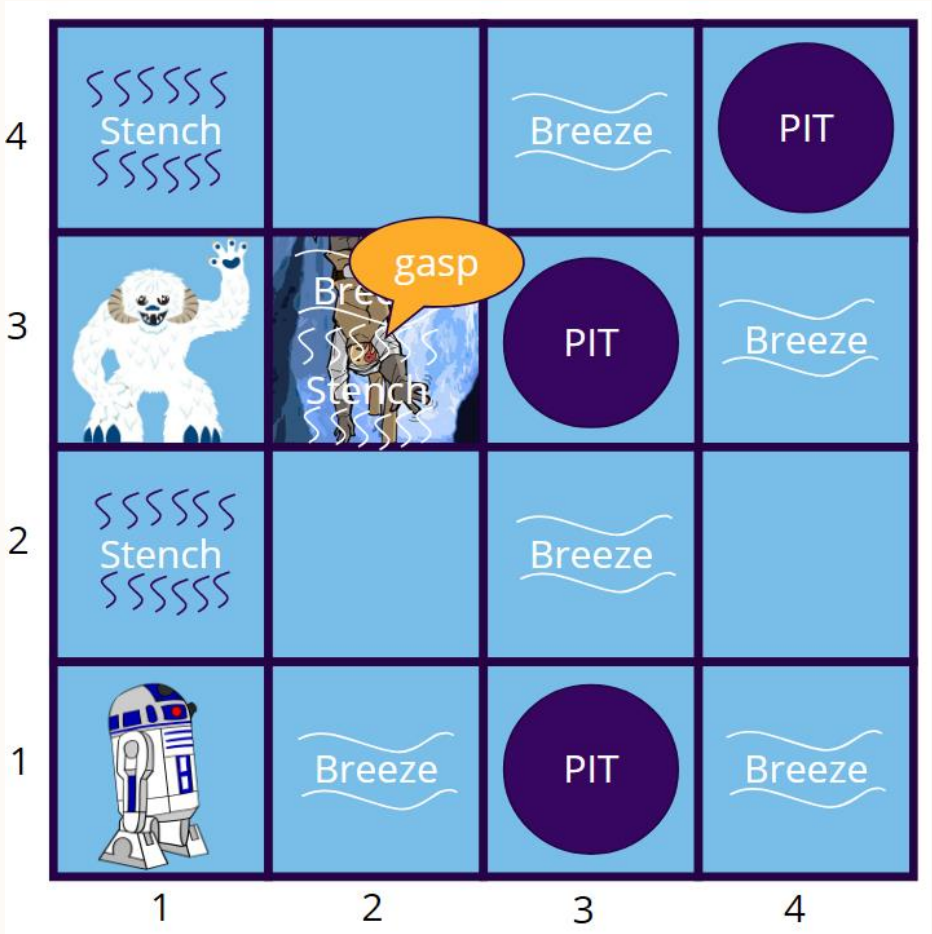
When the Wampa is killed , R2D2's audio sensor perceives a *Scream*

Percept=[Stench, Breeze, Gasp, None, None]



WAMPA WORLD DESCRIPTION

- **Fully Observable?** No – only local perception; location of Luke, Wampa, and pits aren't directly observable
- **Deterministic?**
  Yes – outcomes exactly specified
- **Episodic?** No – sequential at the level of actions; reward isn't given for many steps
  - In an episodic environment, only the current percept is required
  - In a sequential environment, an agent requires memory of past actions to determine the next best actions
- **Static?** Yes – Wampa and Pits do not move
- **Discrete?** Yes
- **Single-agent?** Yes – Wampa is essentially a natural feature

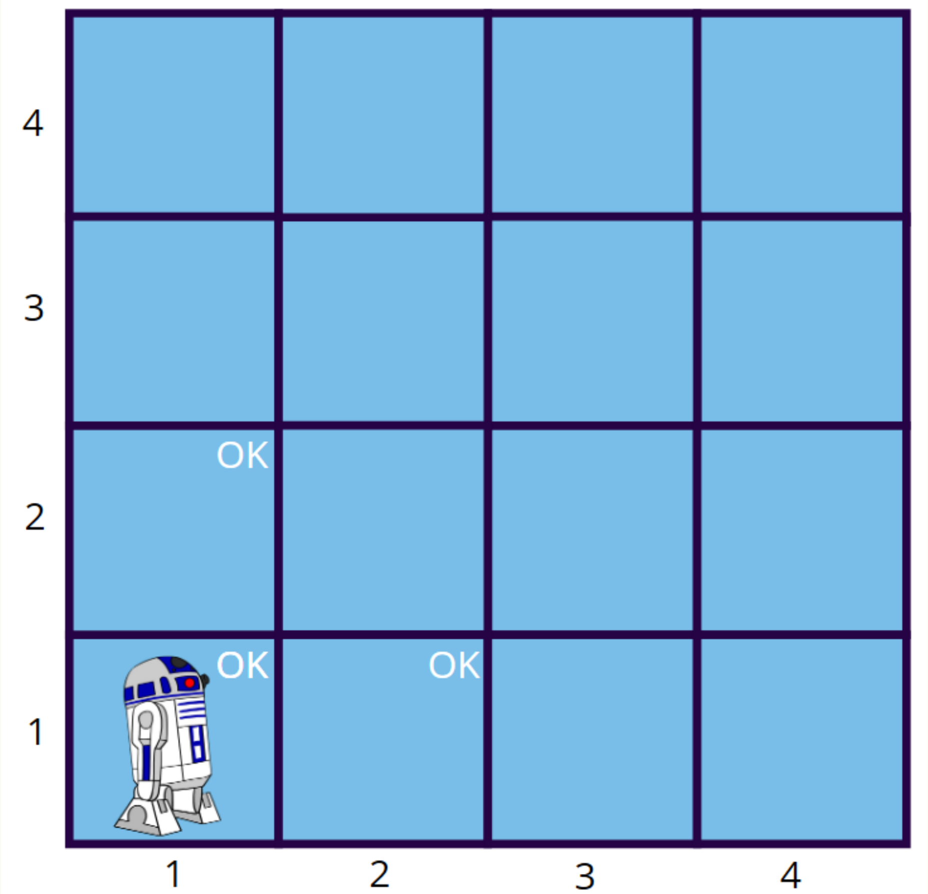


WAMPA WORLD WALKTHROUGH

R2D2 starts in [1,1]

Percept=[$None, None, None, None, None$]

What can we conclude about [1,2] and [2,1]?

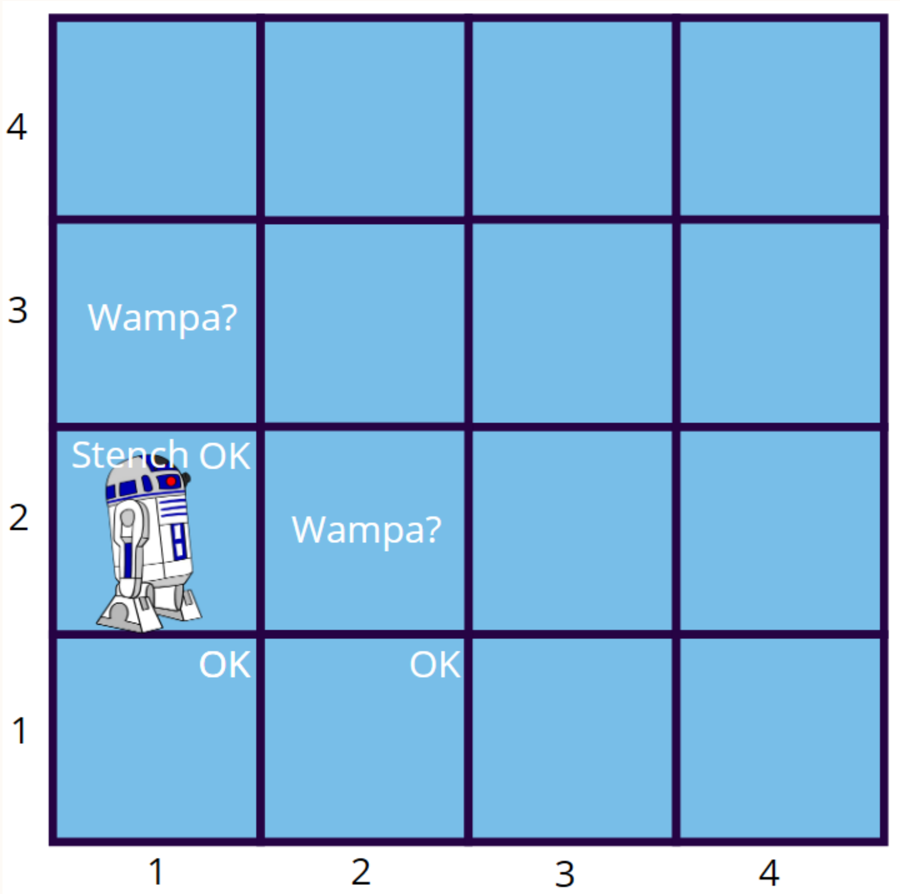


WAMPA WORLD WALKTHROUGH

R2D2 moves to [1,2]

Percept=[$Stench, None, None, None, None$]

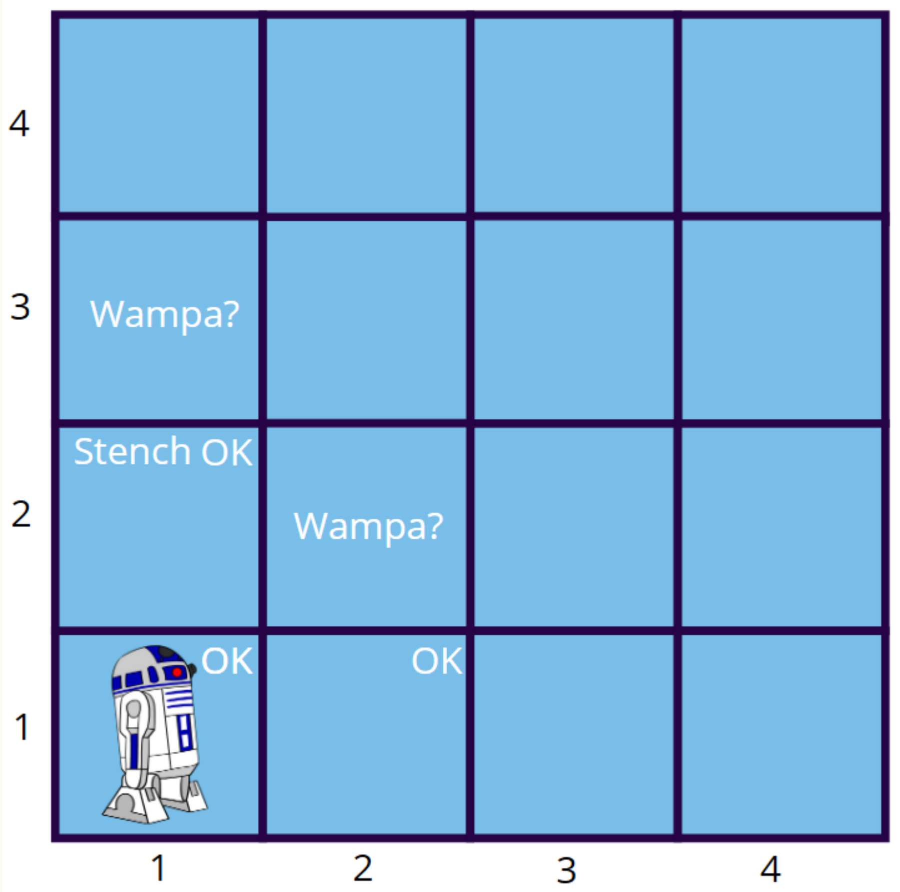
What can we conclude about [3,1] and [2,2] from the *Stench*?



WAMPA WORLD WALKTHROUGH

R2D2 moves back to [1,1] and gets the same percept vector as before

Percept=$[None, None, None, None, None]$

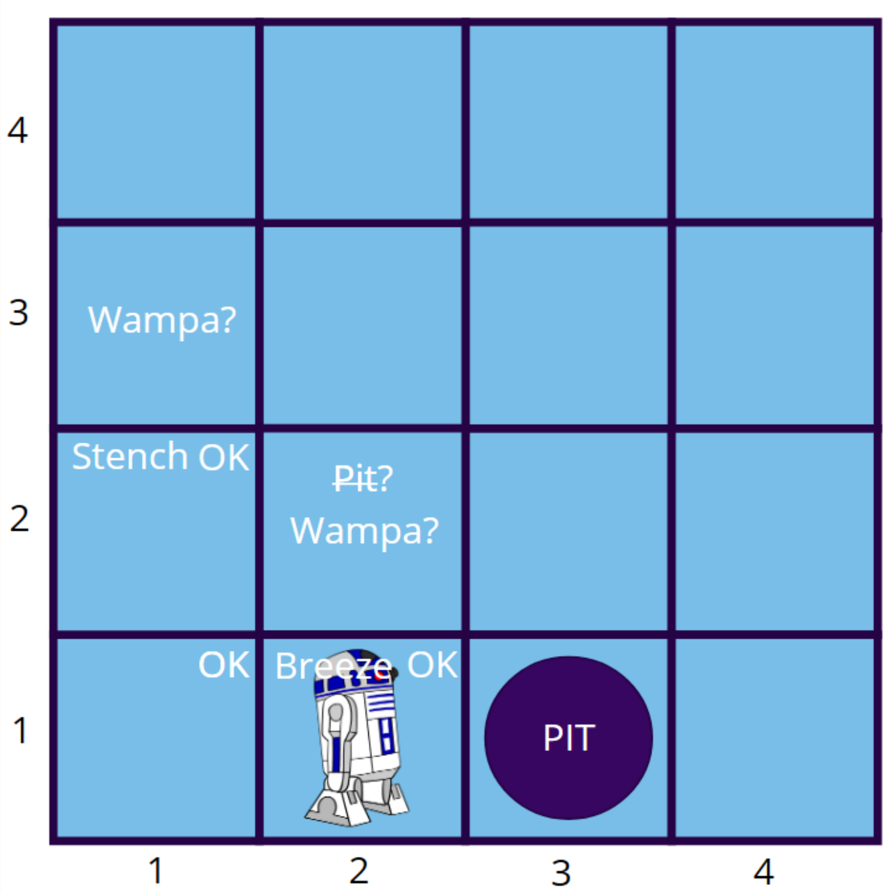


WAMPA WORLD WALKTHROUGH

R2D2 moves to [2,1]

Percept=[*None*, *Breeze*, *None*, *None None*]

What can we conclude about [3,1] and [2,2] based on the *Breeze*?

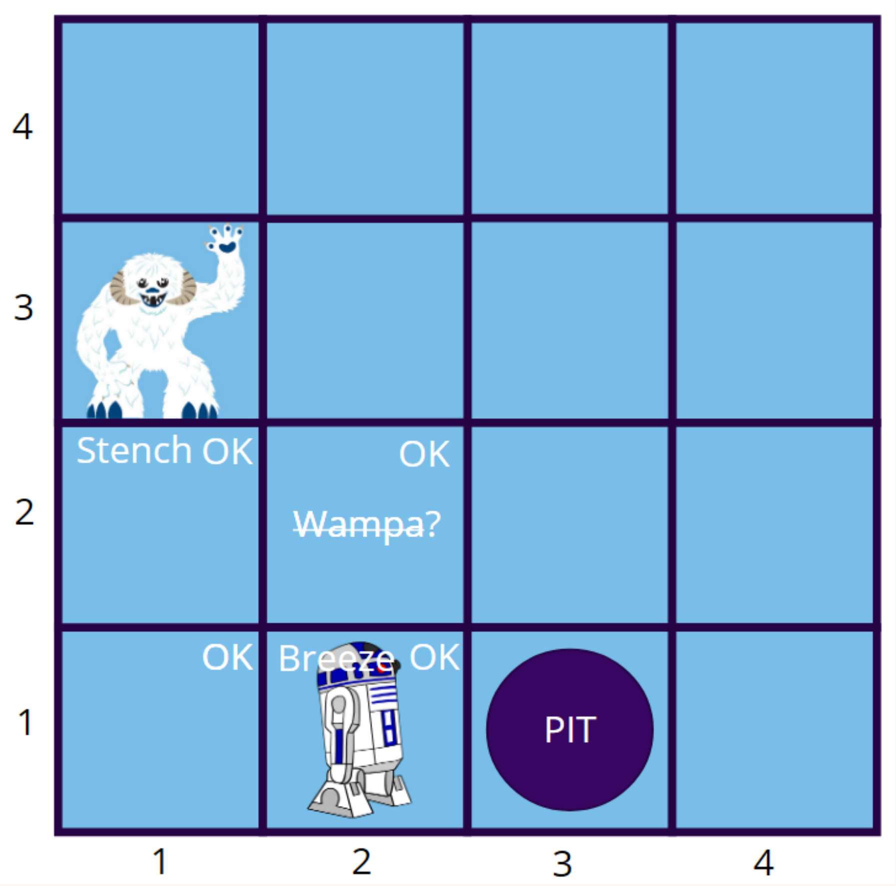


WAMPA WORLD WALKTHROUGH

R2D2 moves to [2,1]

Percept=[*None*, *Breeze*, *None*, *None None*]

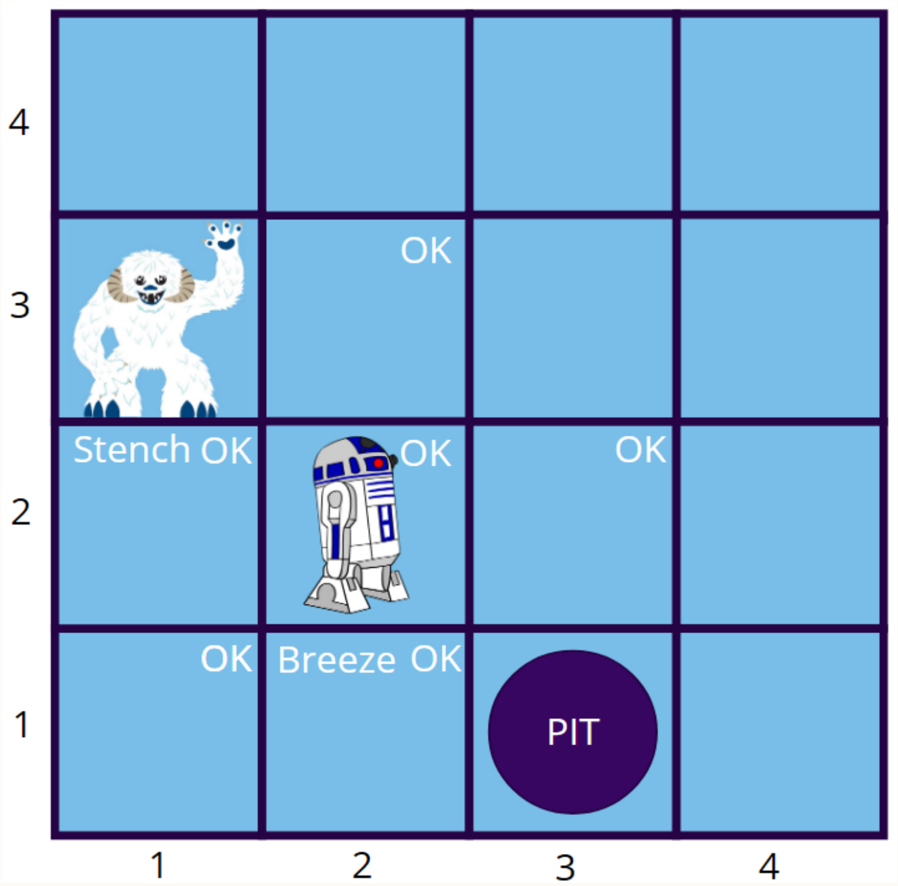
What can we conclude about [2,2] and [1,3] based on the **lack** of a *Stench* here?



WAMPA WORLD WALKTHROUGH

R2D2 moves to [2,2]

Percept=[$None, None, None, None, None$]

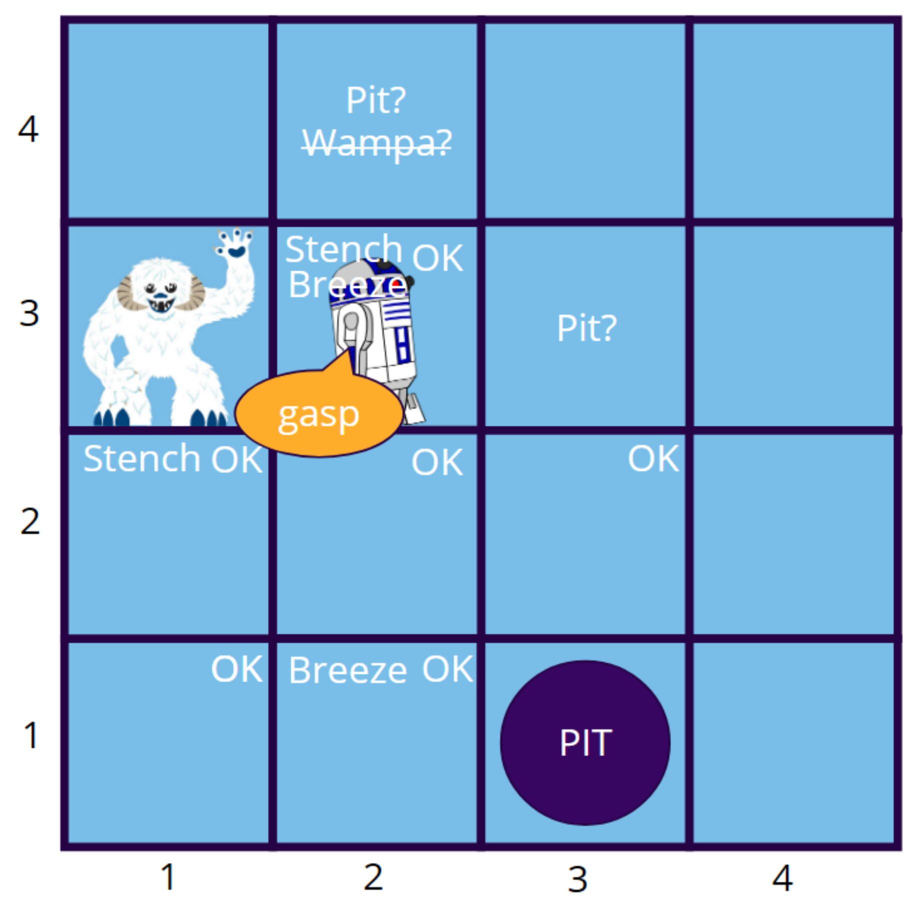


WAMPA WORLD WALKTHROUGH

R2D2 moves to [2,3]

Percept=[*Stench*, *Breeze*, *Gasp*, *None*, *None*]

What can we conclude about [2,4] or [3,3]?

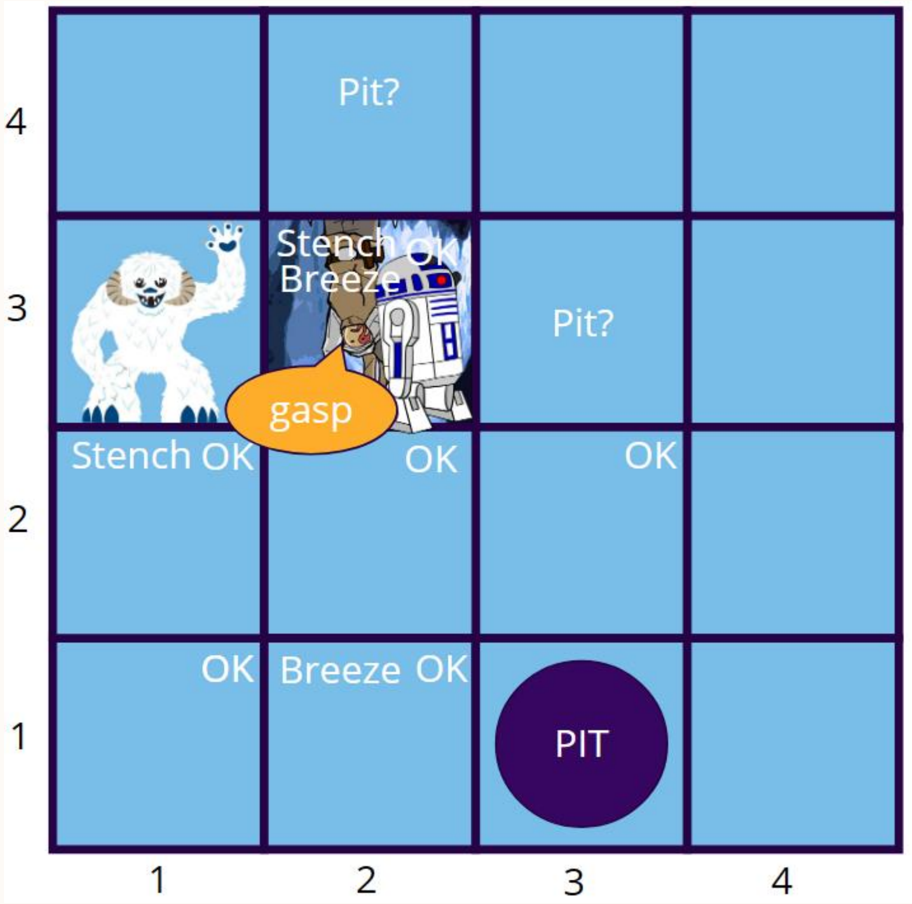


WAMPA WORLD WALKTHROUGH

R2D2 moves to [2,3]

Percept=[*Stench*, *Breeze*, *Gasp*, *None*, *None*]

We heard a *Gasp*, so Luke is here!



HOW DO WE MAKE AN AGENT LIKE THIS?

- Point of knowledge representation is to express knowledge in a **computer usable** form
  - Needed for agents to act on it!
- Knowledge is stored in a Knowledge Base, or KB
- **Logics** are formal languages for representing information such that conclusions can be drawn
- **Syntax** defines how symbols can be put together to form the sentences in the language
- **Semantics** define the "meaning" of sentences;
  - i.e., define **truth** of a sentence in a world (given an interpretation; i.e., a **possible world,** often called a **model**)

POSSIBLE WORLDS/MODELS

- Models are mathematical abstractions that have a fixed set of truth values which are {**true, false**} for each sentence.
- If sentence **α** is true in model **m** then we say
  - **m** satisfies **α**, or
  - **m** is a model of **α**
- We use the notation **M**(**α**) to mean the **set of all models** of **α.**

For instance, **α** could be a sentence that means "there is no pit in [2,2]". In that case, M(**α**) would be all instances of Wampa World where [2,2] doesn't have a pit.

LOGICAL ENTAILMENT

- Once we have a notion of truth, we can start to define **logical reasoning.** Logical reasoning involves the **entailment** relation between sentences.
- Entailment is the idea that a sentence **follows logically** from another sentence.
- To write sentence **α** entails sentence β in mathematical notation we use the **⊨ symbol**:

  $\boldsymbol{\alpha \models \beta}$
- The definition is

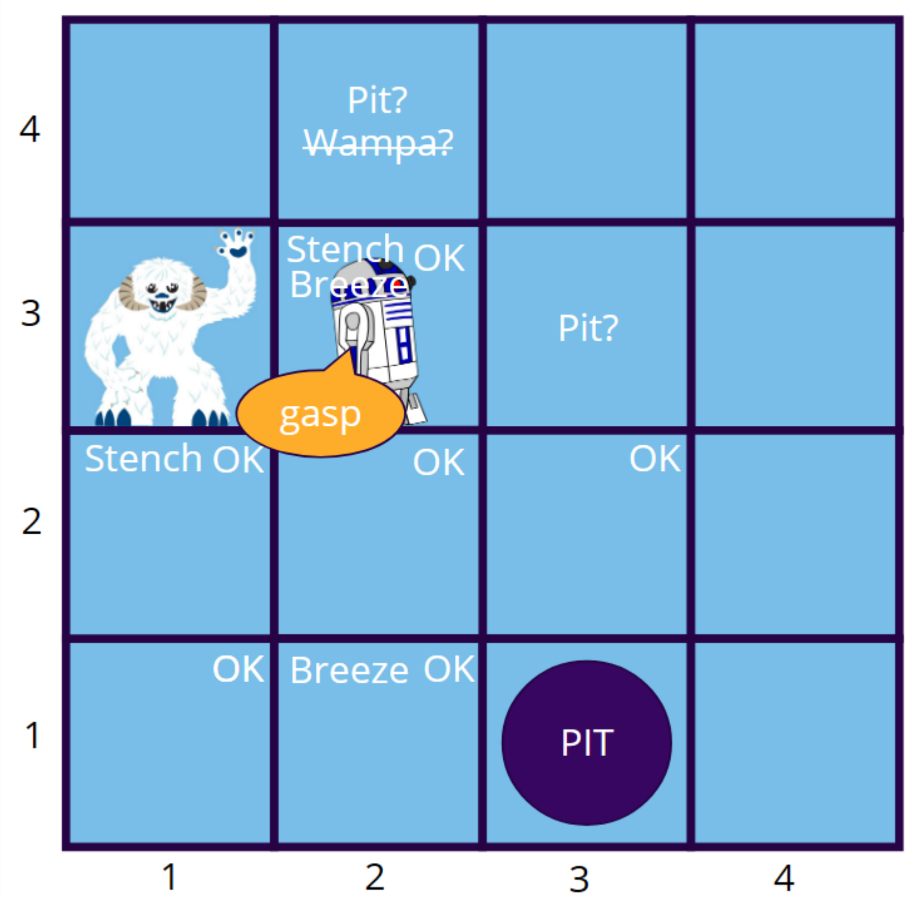
  $\boldsymbol{\alpha \models \beta}$ if and only if (iff) $\mathbf{M(\alpha) \subseteq M(\beta)}$
- This means that **α** is more specific, or stronger than, **β.** For instance, **β** could mean that "The agent is a robot" and **α** could mean "The agent is an astromech".

POSSIBLE WORLDS

- A KB can be thought of as a set of sentences.
  - **α1** = "There is no pit in [1,2]"
  - **α2** = "There is a pit in [3,1]"
  - **α3** = "There is a wampa in [1,3]"
- **The KB is false** in models that contradict what the agent knows. For example, the

KB is false in any model **m** where [1,2] contains a pit.

- **Possible Worlds** is the process of enumerating all Possible Worlds that are compatible with the KB. **M(KB) ⊆ M(α1**)



ENTAILMENT VS DERIVATION

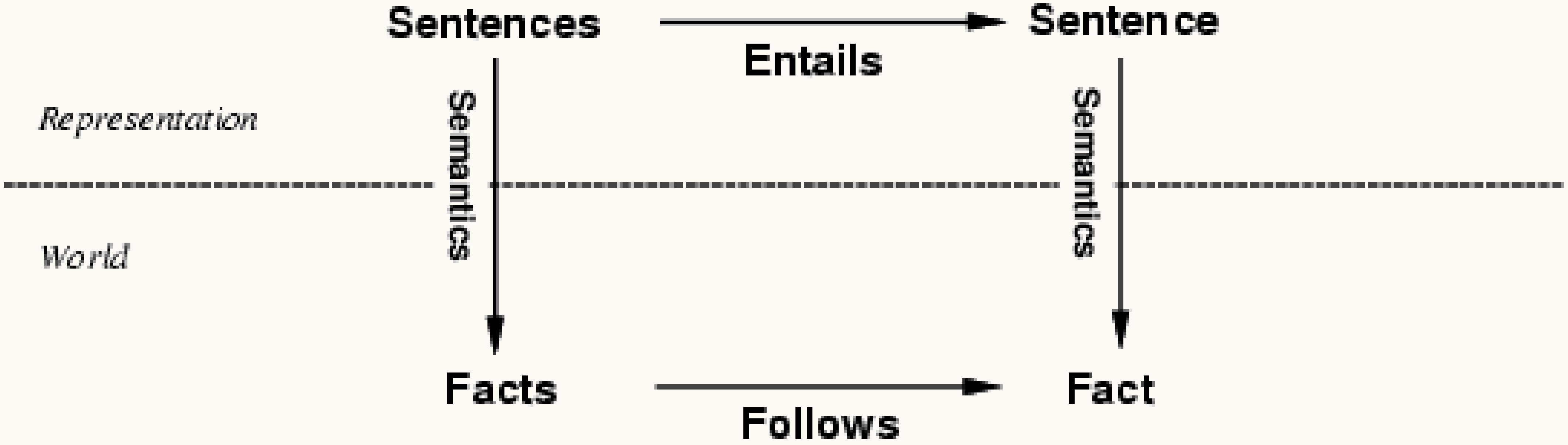
- **Entailment: KB ⊨ Q**
  - Q is entailed by KB (a set of premises or assumptions) if and only if there is no logically possible world in which Q is false while all the premises in KB are true.
  - Or, stated positively, Q is entailed by KB if and only if the conclusion is true in every logically possible world in which all the premises in KB are true.

x ⊨ y: x semantically entails y

- **Derivation: KB ⊢ Q**
  - We can derive Q from KB if there is a **proof** consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

x ⊢ y: y is provable from x

THE 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.

"Dr M is sick with the flu" ⊨ "Dr M is sick"

LOGICAL INFERENCE

- **Inference** is a procedure that allows new sentences to be derived from a knowledge base using an inference algorithm **i.**

  $\mathbf{KB} \vdash_{\mathbf{i}} \alpha$

E.g., from the KB containing "Leia is safe" and "Luke will be happy if Leia is safe," we can infer "Luke is happy" is **true.**

PROPOSITIONAL LOGIC

PROPOSITIONAL LOGIC DEFINITIONS

- **Atomic sentences** are represented with a single propositional symbol
  - **Propositional symbols** stand for a **statement** that can be $\underbrace{\text{true or false}}_{\text{Logical constants}}$.
- For example, $\mathbf{W_{1,3}}$ is a propositional symbol that we choose to stand for "There is a Wampa at location [1,3]"
- That statement can be *true* or *false*.
- The symbol **FacingEast** could stand for "The agent is currently facing East".
  - The user defines the semantics of each propositional symbol

- A **literal** is an atomic sentence or negated atomic sentence.

COMPLEX SENTENCES

- **Complex sentences** are constructed from simpler ones using parentheses and **logical connectives**

Logical Connective	Meaning
$\neg$	Not; $\neg \mathbf{W}_{1,3}$ is the negation of $\mathbf{W}_{1,3}$
$\wedge$	And; $\mathbf{W}_{1,3} \wedge \mathbf{P}_{3,1}$ is called a **conjunction**
$\vee$	Or; $\mathbf{W}_{1,3} \vee \mathbf{P}_{3,1}$ is called a **disjunction**
$\Rightarrow$	Implies; $\mathbf{W}_{1,3} \Rightarrow \mathbf{S}_{1,2}$ is called an **implication**. $\mathbf{W}_{1,3}$ is its **premise** or **antecede** and $\mathbf{S}_{1,2}$ is its **conclusion** or **consequence**
$\Leftrightarrow$	If and only if; $\mathbf{W}_{1,3} \Leftrightarrow \neg \mathbf{W}_{3,4}$ is called a biconditional

TRUTH TABLES

Negation

P	¬P
True	False
False	True

“It is not the case that the Death Star is a moon” is **true** because “the Death Star is a moon” is **false**.

“It is not the case that Wampas smell bad” is **false** because “Wampas smell bad” is **true**.

TRUTH TABLES

Conjunction

P	Q	P ∧ Q
True	True	True
True	False	False
False	True	False
False	False	False

P and Q are both true:
“Wampas smell bad **and** Tauntauns smell bad.” (This sentence is true)

P is true and Q is false:
“Wampas smell bad **and** Tauntauns are robots.” (This sentence is false)

P is false and Q is false:
“Wampas smell good **and** Tauntauns are robots.” (This sentence is false)

TRUTH TABLES

Disjunction

P	Q	P ∨ Q
True	True	True
True	False	True
False	True	True
False	False	False

P and Q are true:
“Wampas smell bad **or** Tauntauns smell bad.” (This sentence is true)

P is true and Q is false:
“Wampas smell bad **or** Tauntauns are robots.” (This sentence is true)

P is false and Q is false:
“Wampas smell good **or** Tauntauns are robots.” (This sentence is false)

TRUTH TABLES

Conditional

P	Q	$P \Rightarrow Q$
True	True	True
True	False	True
False	True	True
False	False	False

To understand why the conditional is defined this way assume that I tell you this $P \Rightarrow Q$:
If you join the dark side then we will rule the galaxy together.
In which of these four scenario did I tell a lie?
1. **You join the dark side**, and **we rule the galaxy together**. (Both P and Q are True)
2. You join the dark side, **but** we **don't** rule the galaxy together. (P is True, Q is False) This one.
3. You **don't** join the dark side, **but** we **still** rule the galaxy together. (P is False, Q is True)
4. You **don't** join the dark side, **and** we **don't** rule the galaxy together. (P is False, Q is False)

TRUTH TABLES

Shorthand for
$P \Longrightarrow Q \land Q \Longrightarrow P$

Biconditional

P	Q	$P \Leftrightarrow Q$
True	True	True
True	False	False
False	True	False
False	False	True

I tell you: The Death Star can be destroyed **if and only if** your missile hits its vulnerable spot.

1. **The Death Star is destroyed**, and **you hit the vulnerable spot**. (Both P and Q are True)
2. The Death Star **is** destroyed, **but** you **didn't** hit its vulnerable spot. (P is True, Q is False)
3. The Death Star **isn't** destroyed, **but** you **did** hit its vulnerable spot. (P is False, Q is True)
4. The Death Star **isn't** destroyed, **and** you **also didn't** hit its vulnerable spot. (P is False, Q is False)

VALIDITY AND SATISFIABILITY

- A sentence is **valid** if it is true in all models
  - E.g.,
    - True
    - $A \lor \neg A$
    - $A \Rightarrow A$
    - $(A \land (A \Rightarrow B)) \Rightarrow B$
- A sentence is **satisfiable** if it is true in some model
  - E.g.,
    - $A \lor B$
    - C
- A sentence is **unsatisfiable** if it is true in no models
  - E.g.,
    - $A \land \neg A$

INFERENCE RULES

- **Logical inference** is used to create new sentences that logically follow from a given set of predicate calculus sentences (KB).
- An inference rule is **sound** if every sentence X produced by an inference rule operating on a KB logically follows from the KB.
  - (That is, the inference rule does not create any contradictions)
- An inference rule is **complete** if it is able to produce every expression that logically follows from (is entailed by) the KB.
  - (Note the analogy to complete search algorithms.)

TWO IMPORTANT PROPERTIES FOR INFERENCE

- **Soundness: If KB $\vdash$ Q then KB $\models$ Q**
  - If Q is derived from a set of sentences KB using a given set of rules of inference, then Q is entailed by KB.
  - Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid.
- **Completeness: If KB $\models$ Q then KB $\vdash$ Q**
  - If Q is entailed by a set of sentences KB, then Q can be derived from KB using the rules of inference.
  - Hence, inference produces all entailments, or all valid sentences can be proved from the premises.

SOUND RULES OF INFERENCE

- Here are some examples of sound rules of inference
  - A rule is sound if its conclusion is true whenever the premise is true
- Each can be shown to be sound using a truth table

RULE	PREMISES	CONCLUSION
Modus Ponens	$A, A \Rightarrow B$	B
And Introduction	A, B	$A \land B$
And Elimination	$A \land B$	A
Double Negation	$\neg \neg A$	A
Unit Resolution	$A \lor B, \neg B$	A
Resolution	$A \lor B, \neg B \lor C$	$A \lor C$
de Morgans	$\neg(A \lor B)$	$\neg A \land \neg B$
$\lor / \Rightarrow$ Equivalence	$A \Rightarrow B$	$\neg A \lor B$

PROVING THINGS

- A proof is a sequence of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference.
- The last sentence is the theorem (also called goal or query) that we want to prove.
- Example for the "weather problem" given above: **Is it raining (R=true), given Hu?**

1. Hu	Premise	"It is humid"
2. $Hu \Rightarrow Ho$	Premise	"If it is humid, it is hot"
3. Ho	Modus Ponens(1,2)	"It is hot"
4. $(Ho \land Hu) \Rightarrow R$	Premise	"If it's hot & humid, it's raining"
5. $Ho \land Hu$	And Introduction(1,3)	"It is hot and humid"
6. R	Modus Ponens(4,5)	"It is raining"

LOGICAL EQUIVALENCE

- Two sentences are logically equivalent iff true in same models: $\alpha \equiv ß$ iff $\alpha \models \beta$ and $\beta \models \alpha$

$(\alpha \land \beta) \equiv (\beta \land \alpha)$	commutativity of $\land$
$(\alpha \lor \beta) \equiv (\beta \lor \alpha)$	commutativity of $\lor$
$((\alpha \land \beta) \land \gamma) \equiv (\alpha \land (\beta \land \gamma))$	associativity of $\land$
$((\alpha \lor \beta) \lor \gamma) \equiv (\alpha \lor (\beta \lor \gamma))$	associativity of $\lor$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \lor \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$	de Morgan
$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$	de Morgan
$(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))$	distributivity of $\land$ over $\lor$
$(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$	distributivity of $\lor$ over $\land$

slide: www.eecis.udel.edu/~mccoy/courses/cisc4-681.10f/lec-materials/Chapt7-7.4+Logical-Agents.ppt

FOR NEXT CLASS

- Continue reading Chapter 7.1-7.5
- Start looking for a paper if you're presenting for Module 2