EECS 496: Sequential Decision Making

Soumya Ray

sray@case.edu

Office: Olin 516

Office hours: T 4-5:30 or by appointment

Today

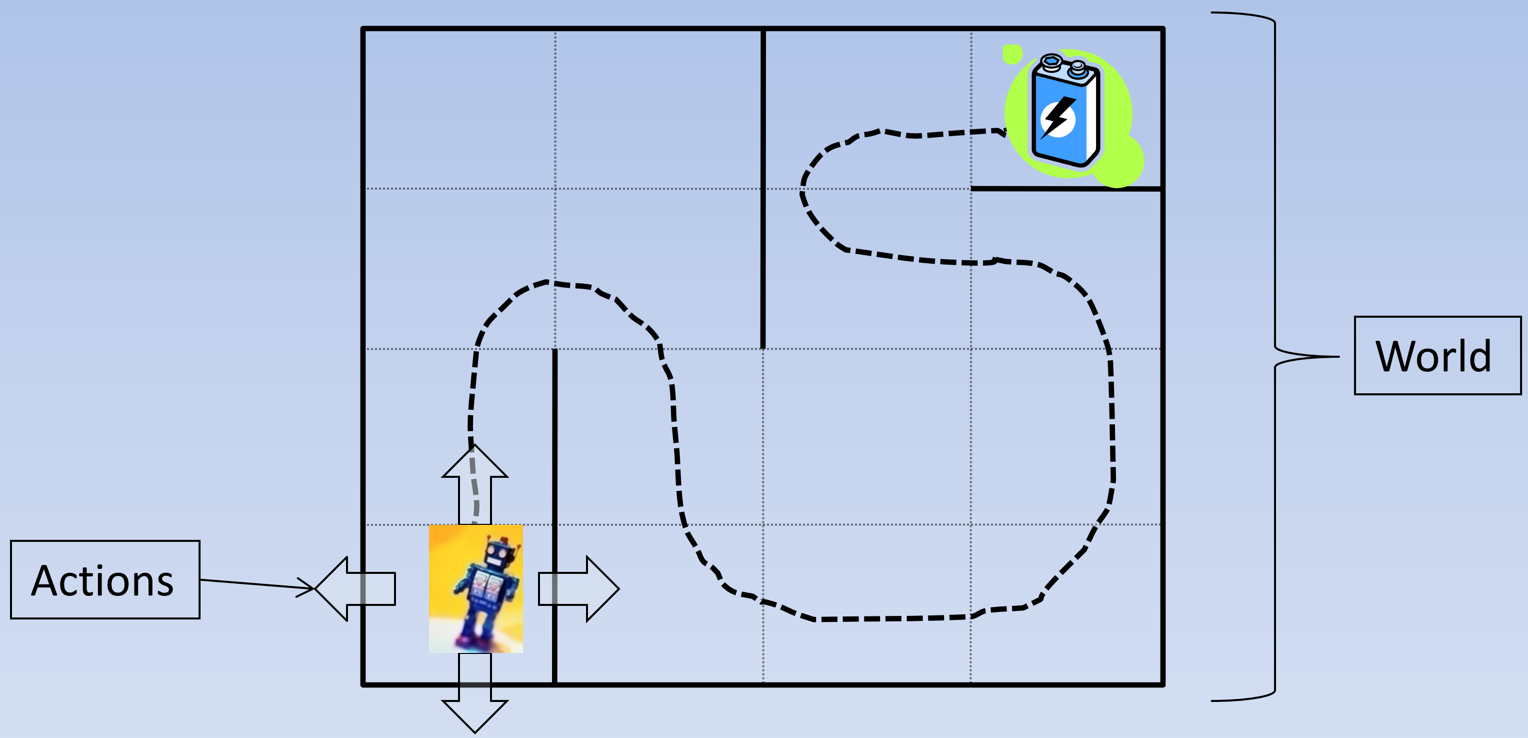
- Propositional Logic (Chapter 7, Russell and Norvig)

What is Sequential Decision Making?

- “Decision Making”: decisions about which *action* to execute in the world

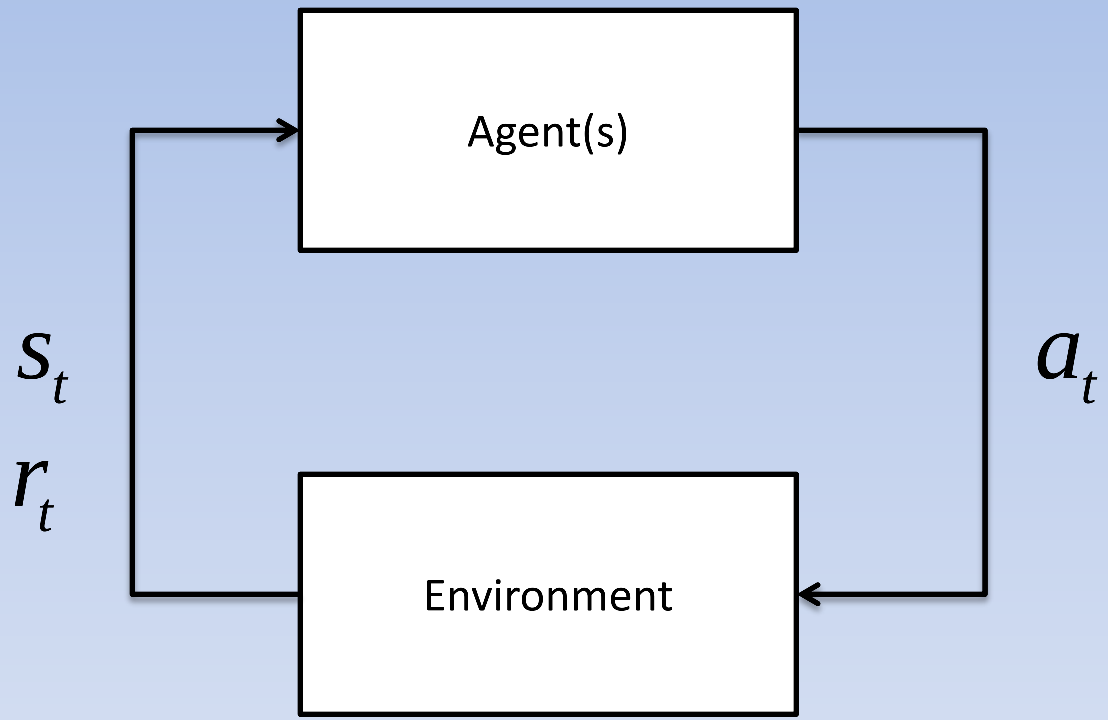
- “Sequential”: Choices made at a given point affect future outcomes and choices

Sequential Decision Making



Find a sequence of actions that maximizes utility over time/ reaches some goal

Agent-Environment Loop



Sequential Decision Making Loop

- Agent starts in some state of the world
- Repeat until done:
  - Agent takes an action based on current (possibly *incomplete, uncertain*) knowledge
  - The state of the world changes as a result (maybe *stochastically)*
  - The world *may* provide some feedback (reward/penalty)
  - The agent gets *(possibly incomplete, uncertain)* knowledge about new state

- "Done": Some sort of terminal or goal state is reached

Representation Language

- To solve an SDM problem, the agent needs to be able to *represent* its environment in its "head" and *reason* with it
  - A *"language"* and associated *inference algorithms*

- The "representational complexity" of an SDM problem depends on the *type* of environment

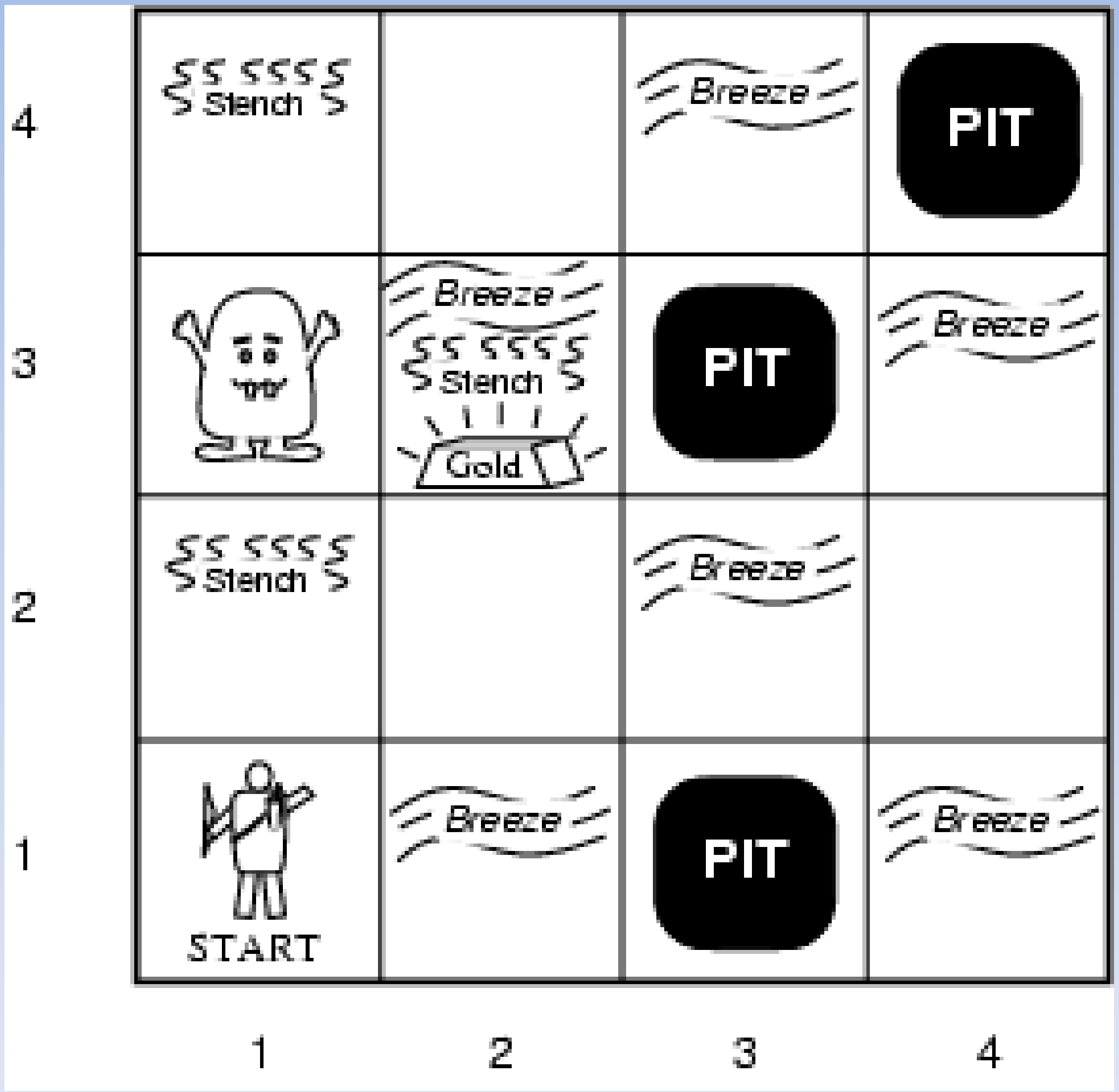
Types of Environments

Type	Definition
Fully observable (vs. *Partially Observable*)	Agent's sensors present complete, accurate picture of the world (as far as determining action sequence is concerned)
Deterministic (vs. *Stochastic*)	The next state of the world is completely determined by current state and agent's action
Static (vs. *Dynamic*)	The world does not change until the agent takes an action
Discrete (vs. *Continuous*)	States, percepts and actions are discrete
Single Agent (vs. *Multiagent*)	The world has only one agent in it
Non-sequential (Episodic) (vs. *Sequential*)	Agent's current action does not affect future actions

Type of Environment

- We'll start by assuming our environments are
  - Deterministic
  - Fully observable
  - Single agent
  - Static
  - Discrete
  - Sequential

Wumpus World



Formalizing the reasoning process

- To build an automated system that can carry out this sort of reasoning, we'll equip them with *knowledge bases* of *facts* and *inference algorithms* to derive new facts

- Propositional logic is a way to formalize the reasoning process described in the previous slide

Syntax and Semantics

- Syntax: what strings of symbols are *allowable sentences* in the language

- Semantics: Given a well-formed sentence, *what does it "mean"*?
  - i.e. how does it connect to the external world?

Propositional Logic Syntax

- A propositional symbol is a "well formed formula" (wff) or "sentence"
  - P, Q, $RainTomorrow$ etc
  - Two special symbols: $TRUE$ and $FALSE$

- Combining any two wffs A and B via the following *logical connectives* results in a wff:
  - $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$, $(A \Leftrightarrow B)$

Valid or not?

- $(A \wedge B) \Rightarrow (A \vee B)$ ✓
- $(((A \Leftrightarrow B) \Leftrightarrow C) \Leftrightarrow D)$ ✓
- $(\neg A \Rightarrow B \vee C) \wedge (A \Leftrightarrow B)$ ✗
- $(A \neg \Rightarrow B)$ ✗

Note on syntax

- Do not confuse the English words “and”/ “or” etc with the logical connectives

Semantics: Models

- A "model" assigns truth values (true or false) to all propositional symbols in the language
  - Every model assigns a truth value "true" to $TRUE$
  - No model assigns a truth value "true" to $FALSE$

Semantics of Logical Connectives

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

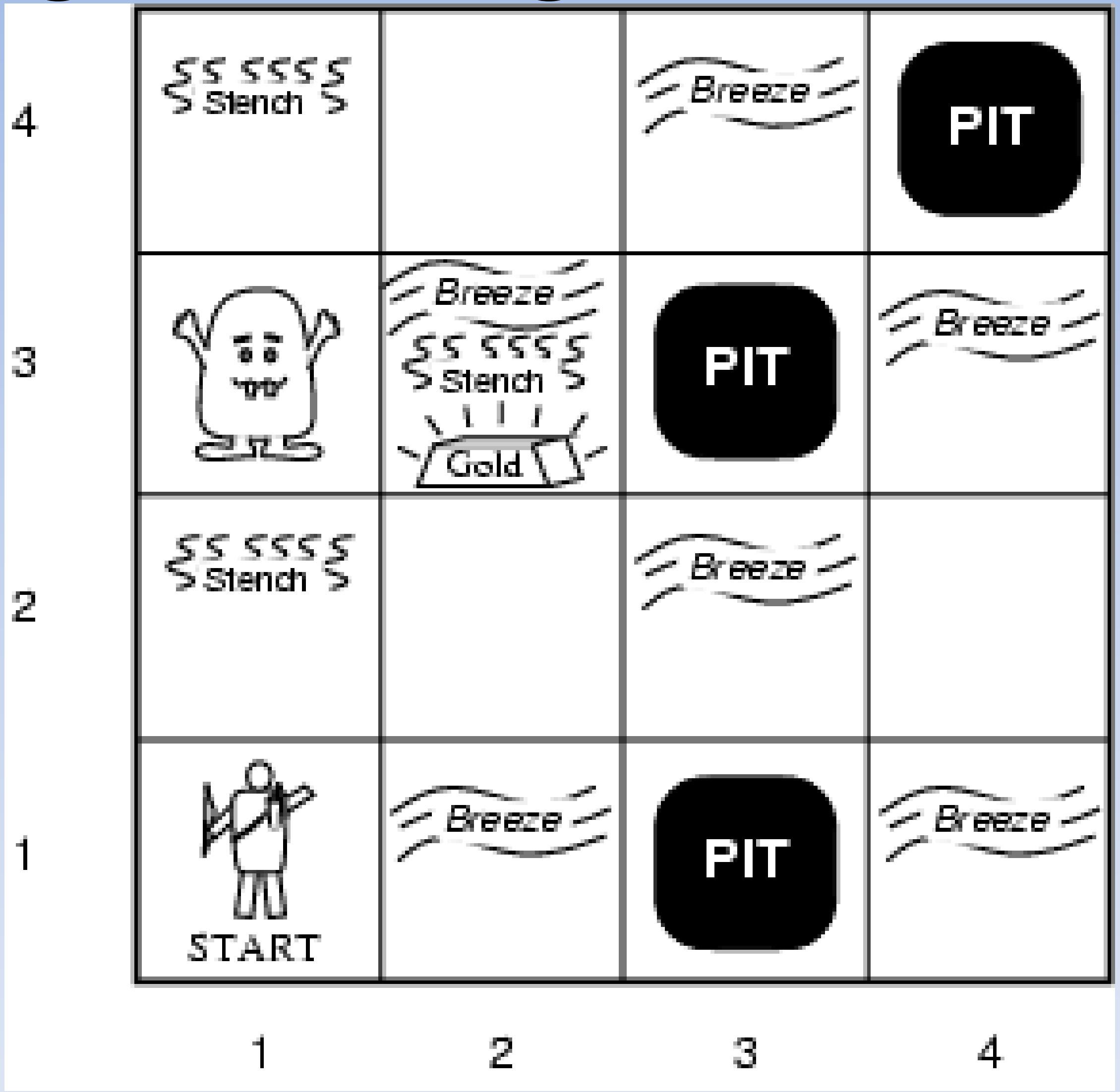
Truth Tables

Semantics of sentences

- The semantics defines the truth value of a sentence (wff) with respect to the model

- We say a model M is "a model of a sentence α" if α evaluates to "true" when the symbols in α are assigned values according to M

Building Knowledge Bases



Building Knowledge Bases

- Let $P_{i,j}$ represent the existence of a pit in [i, j].
- Let $B_{i,j}$ represent the existence of a breeze in [i, j]. Similarly for $S_{i,j}$
- Agent knows: $\neg P_{1,1} \wedge \neg P_{1,2} \wedge \neg P_{2,1} \wedge \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1} \wedge \neg B_{1,1} \wedge B_{2,1} \wedge \neg B_{1,2} \wedge \neg S_{1,1} \wedge S_{1,2} \wedge \neg S_{2,1}$
- "Pits cause breezes in adjacent squares"

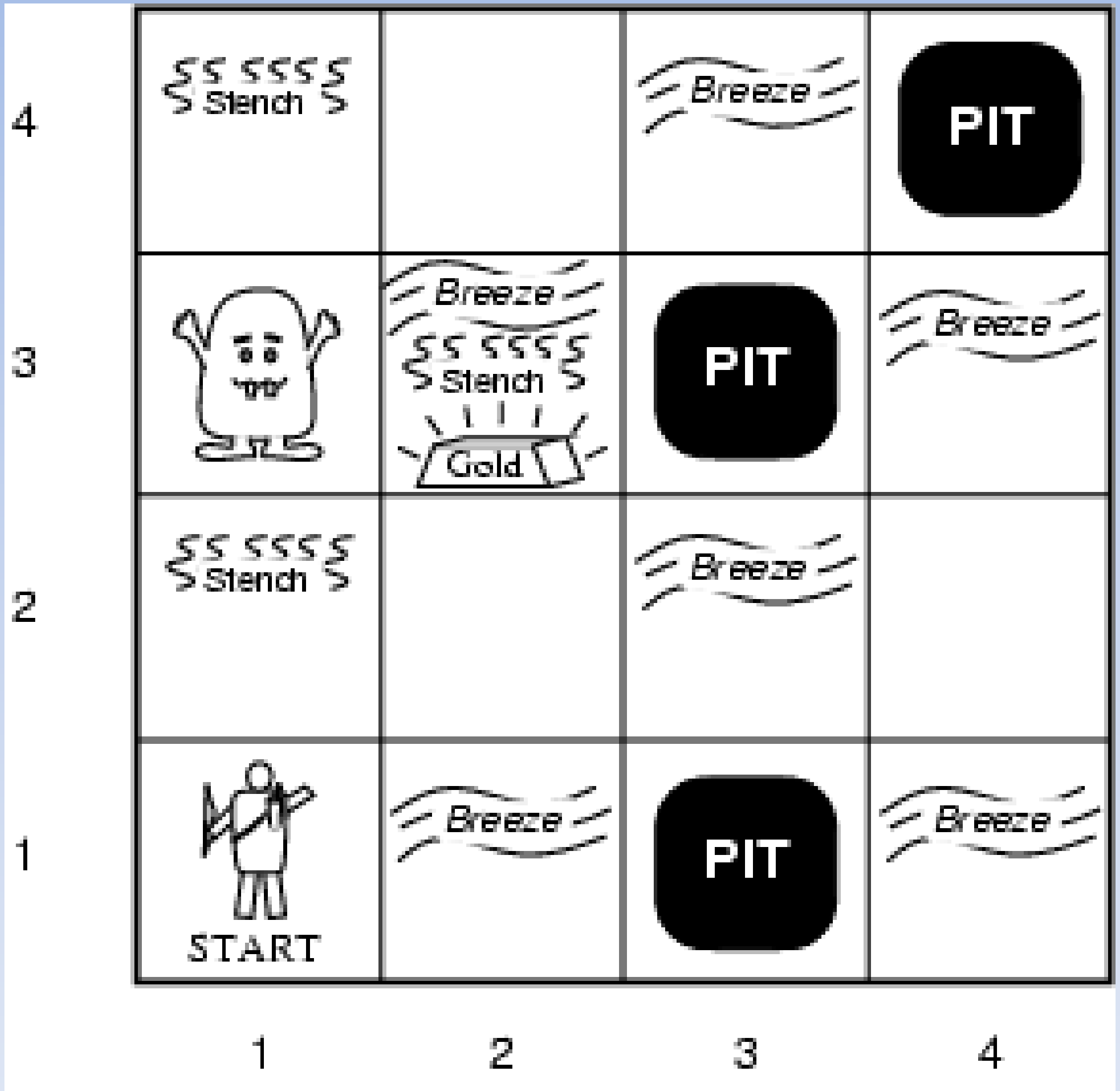
$$(P_{1,1} \Rightarrow (B_{1,2} \wedge B_{2,1})) \wedge$$
$$(P_{2,1} \Rightarrow (B_{1,1} \wedge B_{2,2} \wedge B_{3,1})) \wedge \ldots$$

Entailment

- Given a knowledge base, we will need to identify what else is a consequence of this
  - This is *logical inference (deduction)*

- Inference in logic involves establishing an *entailment* relationship: $\alpha \models \beta$
  - $\alpha \models \beta$ (α entails β) iff every model of α is also a model of β

Wumpus World



Suppose our knowledge base has facts “Breeze in 2,1” and “Nothing in 1,1”

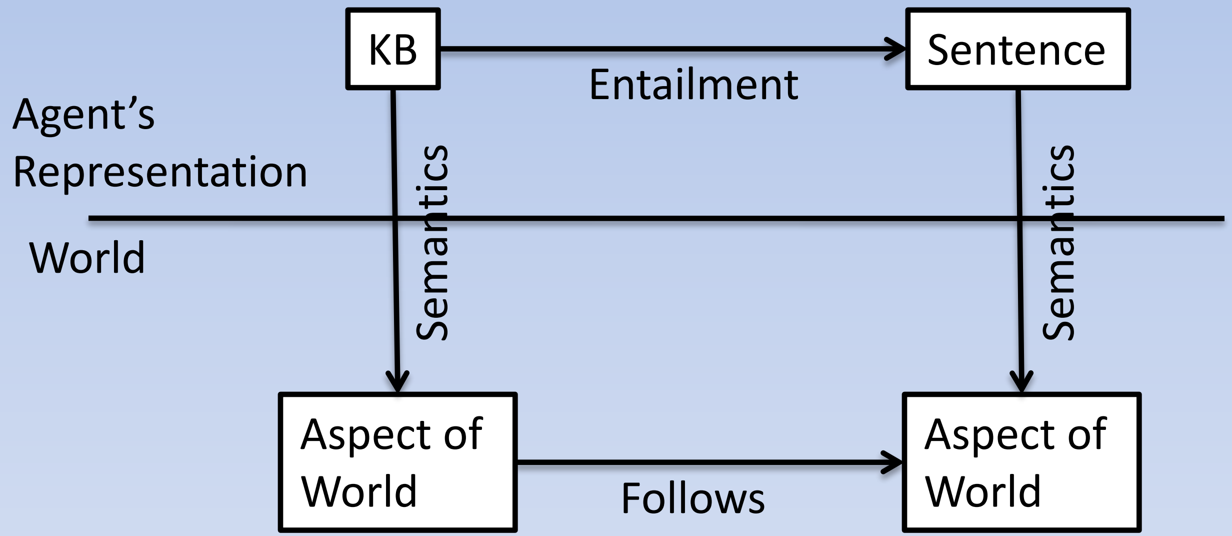
This KB entails “Nothing in 2,1” but does not entail “Nothing in 2,2”

Note: If α does not entail β then it does NOT mean $\alpha \models \neg\beta$

Note on syntax

- Things like ($\alpha \models \beta \wedge \gamma \models \delta$) are NOT wffs
  - $\models$ is a "meta-logical" symbol
  - It says something ABOUT the logic but is not IN the logic
  - Often $\models$ is confused with $\Rightarrow$
  - $((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ is a wff
  - "$\alpha \models \beta$ and $\gamma \models \delta$" is also ok
    - Shorthand for "every model of α is a model of β *and* every model of γ is a model of δ"

Putting things together



Inference by Enumeration

- Given a propositional knowledge base (KB), how will the agent deduce new facts?

- For a given fact α, does KB $\models \alpha$?
  - i.e., every model of KB is a model of α
  - Enumerate all possible models and check

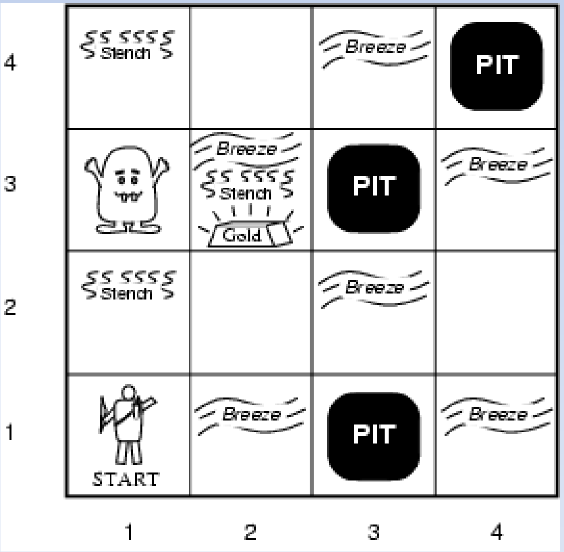
Inference by Enumeration

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
$false$	$false$	$false$	$false$	$false$	$false$	$false$	$false$	$true$
$false$	$false$	$false$	$false$	$false$	$false$	$true$	$false$	$true$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$false$	$true$	$false$	$false$	$false$	$false$	$false$	$false$	$true$
$false$	$true$	$false$	$false$	$false$	$false$	$true$	$\underline{true}$	$\underline{true}$
$false$	$true$	$false$	$false$	$false$	$true$	$false$	$\underline{true}$	$\underline{true}$
$false$	$true$	$false$	$false$	$false$	$true$	$true$	$\underline{true}$	$\underline{true}$
$false$	$true$	$false$	$false$	$true$	$false$	$false$	$false$	$true$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$true$	$true$	$true$	$true$	$true$	$true$	$true$	$false$	$false$

Unfortunately, in the worst case, no more efficient strategy is possible.

Inference by Enumeration

B_{12}	B_{21}	S_{12}	S_{21}	P_{22}	$P_{22} \Rightarrow B_{12} \wedge B_{21}$	$\neg P_{22}$
T	F	F	T	F	T	T
T	F	F	T	T	F	F



Can we do better?

Definitions

- A formula α is valid if it is true in every model (tautology)

- A formula α is satisfiable if it is true in *some* model

Satisfiability and Inference

- Satisfiability and inference are linked by the proof-by-refutation theorem: $\alpha \models \beta$ iff $(\alpha \wedge \neg\beta)$ has no model (is unsatisfiable)