CS 5100: Foundations of Artificial Intelligence

Agents, Logic, and Reasoning

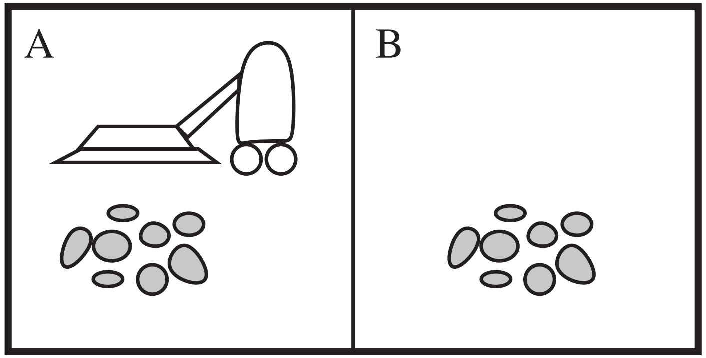
Prof. Amy Sliva

September 15, 2011

Outline

- Agents review
- Knowledge representation and engineering
- Formal logic
- Reasoning with propositional logic
- Discuss project 2

Vacuum-cleaner world



- Percepts: location and contents, e.g., [A,Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*
- Production rules:

  [A, Dirty] → *Suck*
  [B, Dirty] → *Suck*
  [A, Clean] → *Right*
  [B, Clean] → *Left*

An agent's abilities

- Consider the **reflex** vacuum agent
  - Lacks explicit knowledge of the environment
  - Lacks memory of its own past behavior
  - Lacks knowledge of the effects of what it does
- The environment may **behave reasonably**, but the agent does not have any machinery for understanding that concept or making predictions
- **Note:** what do we mean by behave reasonably???
  - According to **OUR COMMON SENSE!!!**

“Reasonable” behavior of the environment

1. If dirty and action ≠ suck, that square will be dirty the next time it is perceived
2. If action is suck or nop then the next percept is the same location
3. If action is left or right then the next percept is the expected location
   (note: what if we are at B and go right??)

- What kind of agent could include this kind of **common sense knowledge** and take advantage of it?

Let’s design some agents

- Environment with three squares: A, B, C and the same actions and percepts as vacuum world
  - How would you design a reflex agent? (i.e., specify the production rules)
- What about a grid environment?
  - Need additional actions (let’s say N, S, E, W)
- **Declarative representation** of history: [[loc, action], ...]
  - Less efficient, but more flexible than **special-purpose design** (e.g., array where each element represents a square and contains the time step if its most recent cleaning)

Evaluating an agent's performance

- Consider a vacuum agent trying to be rational under a performance measure that assigns a cost to actions and a reward for keeping the room clean
  - How to represent this knowledge ? How it use it?
  - Would it need knowledge of its own prior actions (?)

Consider a vacuum agent with parameters:

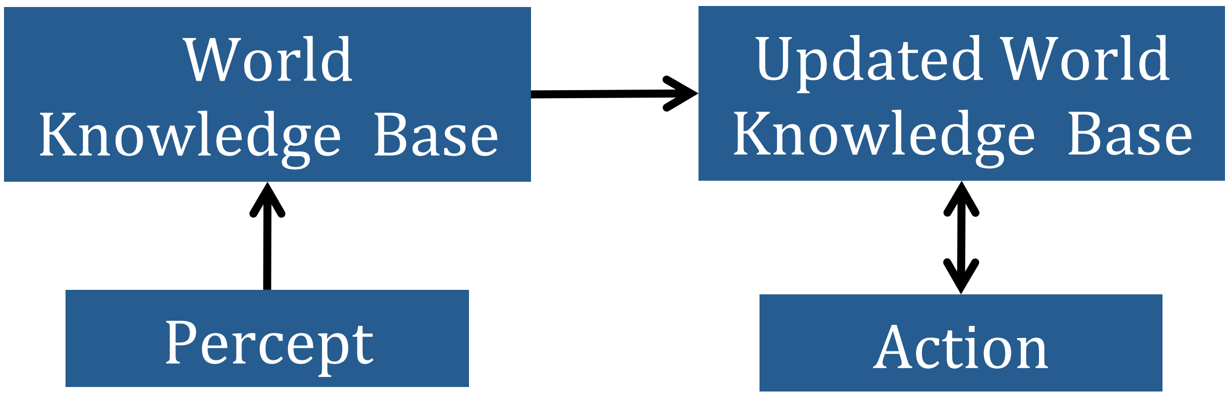
C_i = cost of each action i

P = penalty for each dirty square at each time step

Strategy to minimize total cost + total penalties

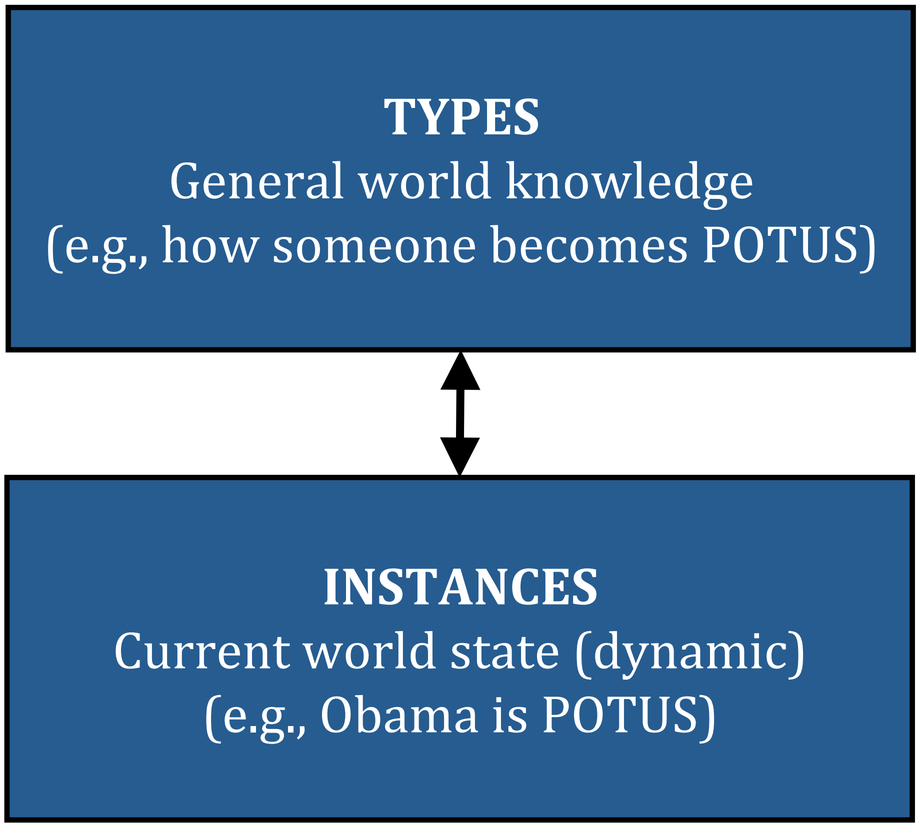
NOTE: what else would we want to know ?????

Elements of a knowledge-based agent



- Three main components
  1. A formal language for expressing knowledge declaratively
     - **Knowledge representation**
  2. A knowledge base design to express what is known
     - **Knowledge engineering** or **ontology**
  3. Algorithms to use and update the knowledge base
     - **Automated inference**

Ontology design ("knowledge engineering")



New percepts are added to **instances** box

Example

World Knowledge Base

TYPES include family relations such as:

- a female with the same parents is a sister
- a male with the same parents is a brother
- parents of your parents are your grandparents
- a male child of your brother or sister is your nephew

INSTANCES includes: Sam and Mary have a male child named Max

↑

New Percept: Sam and Mary have a female child named Sarah

Updated World Knowledge Base

Sam and Mary have a female child named Sarah
Max has a sister named Sarah
Sarah has a brother named Max

Percept: Mary has a father named Tom -- ??
Percept: Max has a son named Simon -- ??

Knowledge representation using formal logic

- The **declarative** approach: beliefs are represented as a set of **sentences** in a formal logic
- The sentences are believed to be true by the agent
- Our first representation language: **propositional logic**
  - Simple and easy to compute but not very expressive
- Trade-off in choosing a representation language
  - Expressiveness v. Tractability
  - Horn clauses $\subset$ logic sentences

Limitations of logic to represent knowledge

- **Beliefs** of an agent—"understand" and reason about the current world state
  - Logic is powerful and useful, but requires agent beliefs to be definite (True/False) and consistent
- **Dynamic/causal** knowledge—"understand" and reason about how the world changes
  - Temporal logic is quite awkward
  - Frame problem
- Deciding how to act (**planning**)
  - Represent the current state and goal state with logic
  - Aspects of real-world planning (dynamics, uncertainty, concurrent events, multiple agents, competing goals) not well suited for logic
- **Human behavior**—reasoning about motivations, preferences, emotions, cooperation, and competition
  - Not well suited for logic

Formal logic basis of automated reasoning

- If beliefs include both ***p*** and ***p*** → ***q***, then the agent can infer ***q***. (This logical inference rule is called ***modus ponens***)

- **Example:**
  - B1. raining → ground is wet
  - P1 → raining

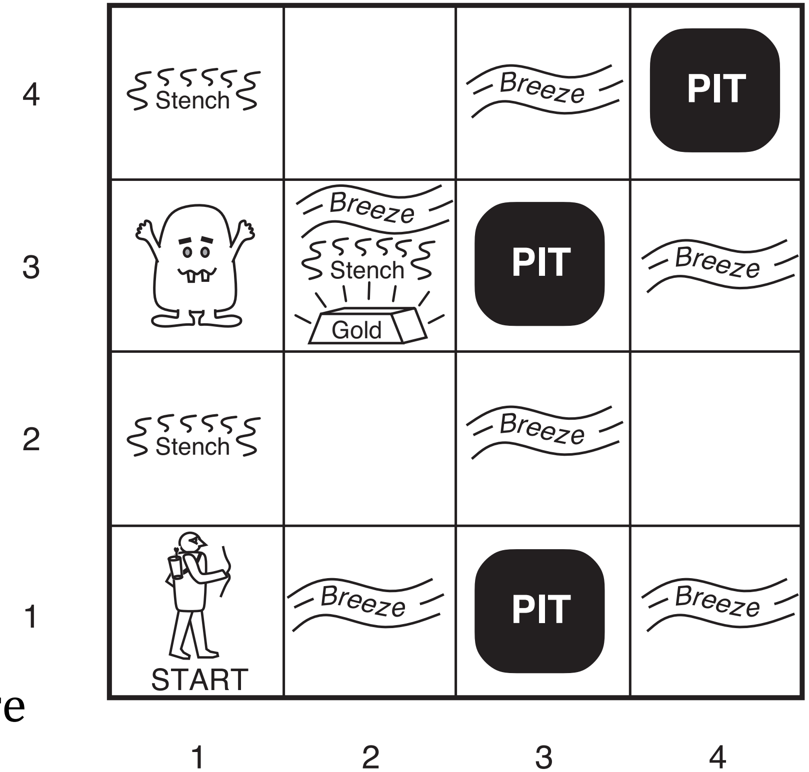
  Infer: ground is wet
  - B2. ground is wet → ground is slippery

  Infer: ground is slippery

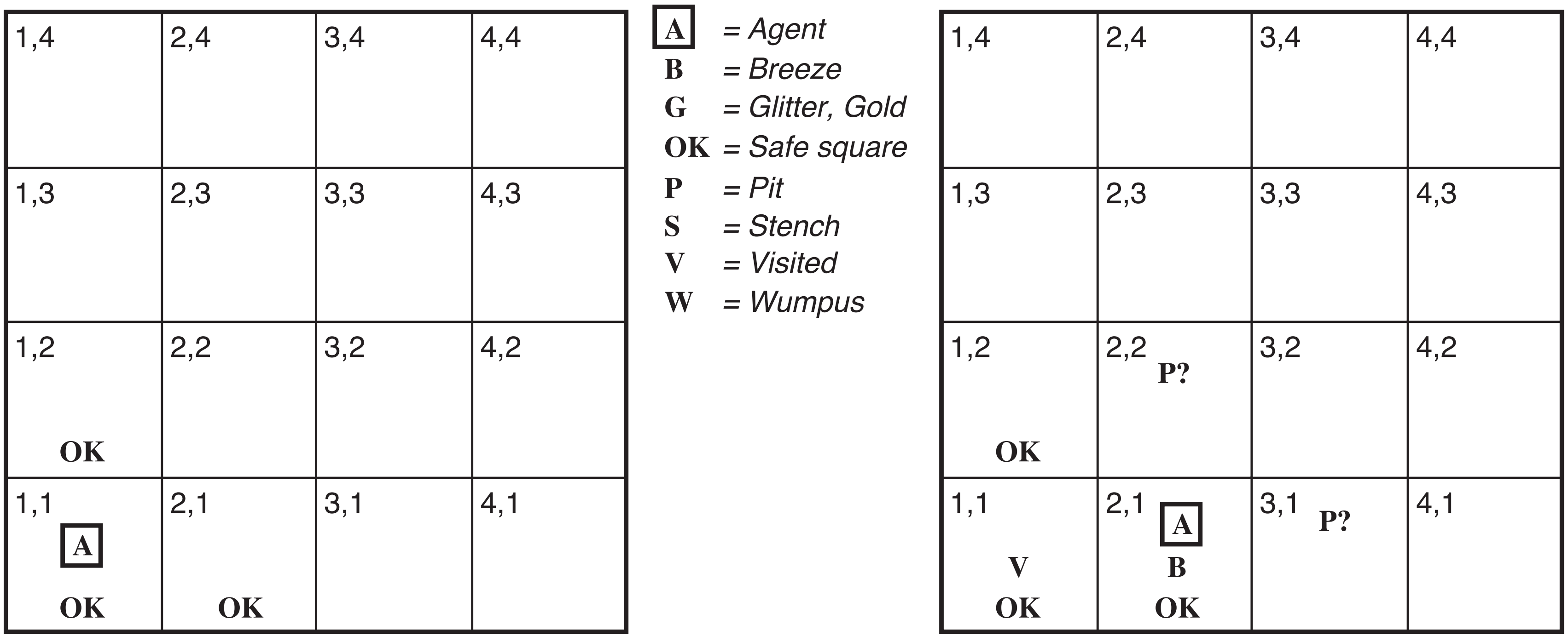
- This is called "chaining" and by chaining we can produce complex reasoning sequences

Wumpus World PEAS description

- **Performance measure**
  - gold = +1000, death = -1000
  - -1 per step, -10 for using the arrow
- **Environment**
  - 4x4 grid—agent starts at [1,1]
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to a pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if it in same square
  - Climbing exits the cave if at [1,1]
- **Actuators:** *Forward, TurnLeft, TurnRight, Grab, Shoot, Climb*
- **Sensors:** Stench, Breeze, Glitter, Bump, Scream



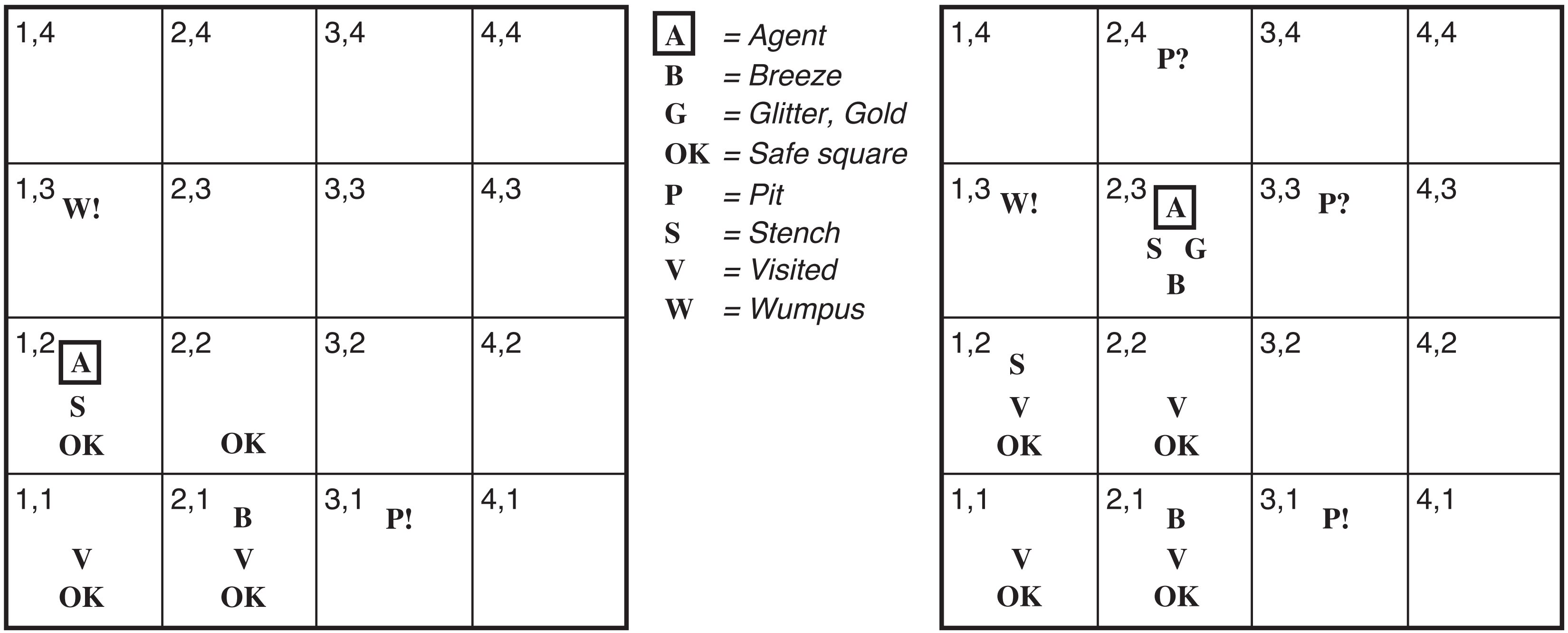
Exploring a wumpus world



(a) (b)

1. [None, None, None, None, None]→Forward to [2,1]
2. [None, Breeze, None, None, None]→ Back to [1,1], then to [1,2]
   - Possible pit in [2,2] or [3,1] or both

Exploring a wumpus world



(a) (b)

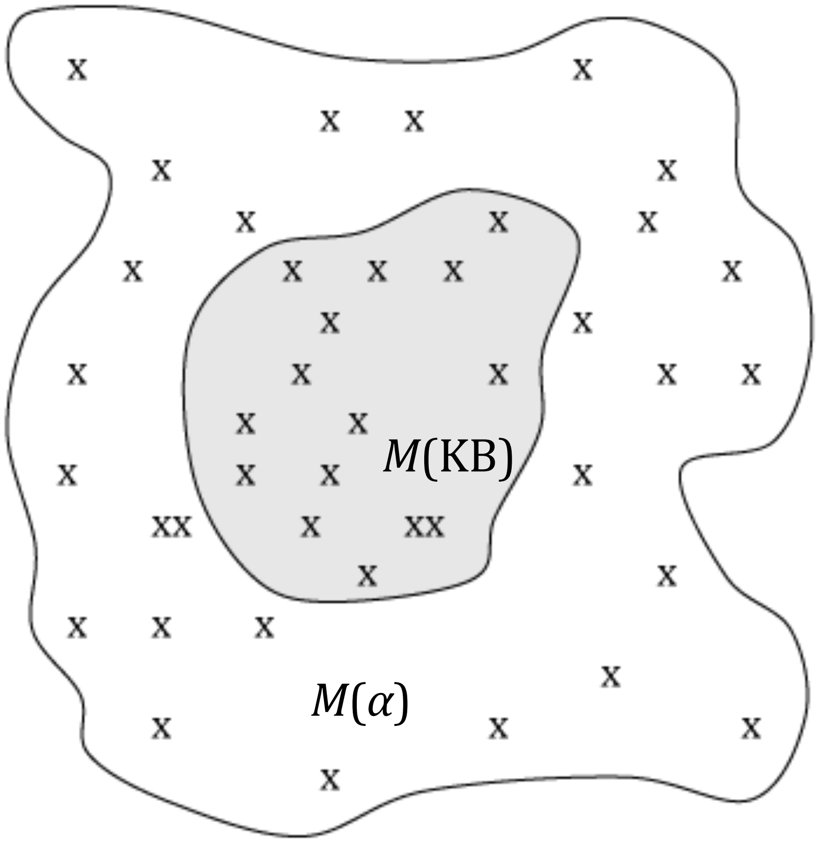
1. [Stench, None, None, None, None]→ Move to [2,2]
   - Wumpus must be in [1,3]!
   - No breeze, so no pit in [2,2]—must be at [3,1]
2. [...Percepts...]→ Move to [2,3]
3. [Stench, Breeze, Glitter, None, None]

Logic in general

- **Logics** are formal languages for representing information such that conclusions can be drawn
- **Syntax** defines the acceptable sentences in the language
  - Well formed formulas ("wffs")
- **Semantics** define the "meaning" of sentences
  - How to decide the **truth** of a sentence in a world model
- E.g., the language of arithmetic
  - **`x + 2 > y`** is a sentence; **`x2 + y >`** is not
  - `x + 2 > y` is true iff the number `x + 2` is greater than the number `y`
  - `x + 2 > y` is **`true`** in a world where `x = 7`, `y = 1`
  - `x + 2 > y` is **`false`** in a world where `x = 1, y = 7`

Models

- Semantics for logic is **truth-functional** and defined in term of **models**
  - Models—structured worlds w.r.t. which truth can be evaluated
  - If m is a model, then it assigns true or false to every expression
- We say m is a model of a sentence α if α is true in m
- $M(\alpha)$ is the set of all models for α
- Then $\text{KB} \vDash \alpha$ iff $M(\text{KB}) \subseteq M(\alpha)$
  - E.g., KB = Giants won and Reds won
    α = Giants won



Entailment

- **Entailment** means that a sentence **follows logically** from another
  - KB $\models \alpha$
- Knowledge base KB **entails** sentence α iff α is true in **all** worlds where KB is true
  - $M(\text{KB}) \subseteq M(\alpha)$
  - E.g., the KB containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"
  - E.g., x + y = 4 entails 4 = x + y
- Entailment is a relationship between sentences (i.e., **syntax**) that must also match the **semantics**
- **Remember:** KB is a database of what the agent knows

Example: semantics of $M(\alpha)$

- $M(\alpha)$ is the **set of all models** of α
- Possible worlds

$\mathbf{W_1}$	$\mathbf{W_2}$	$\mathbf{W_3}$	$\mathbf{W_4}$
Giants won Reds won	Giants won	Reds won	

- What worlds are included in each of these:
  - $M(\textit{Giants won})$
  - $M(\textit{Reds won})$
  - $M(\textit{Giants won and Reds won})$
  - $M(\textit{Giants won or Reds won})$

Example: world models

- Possible worlds

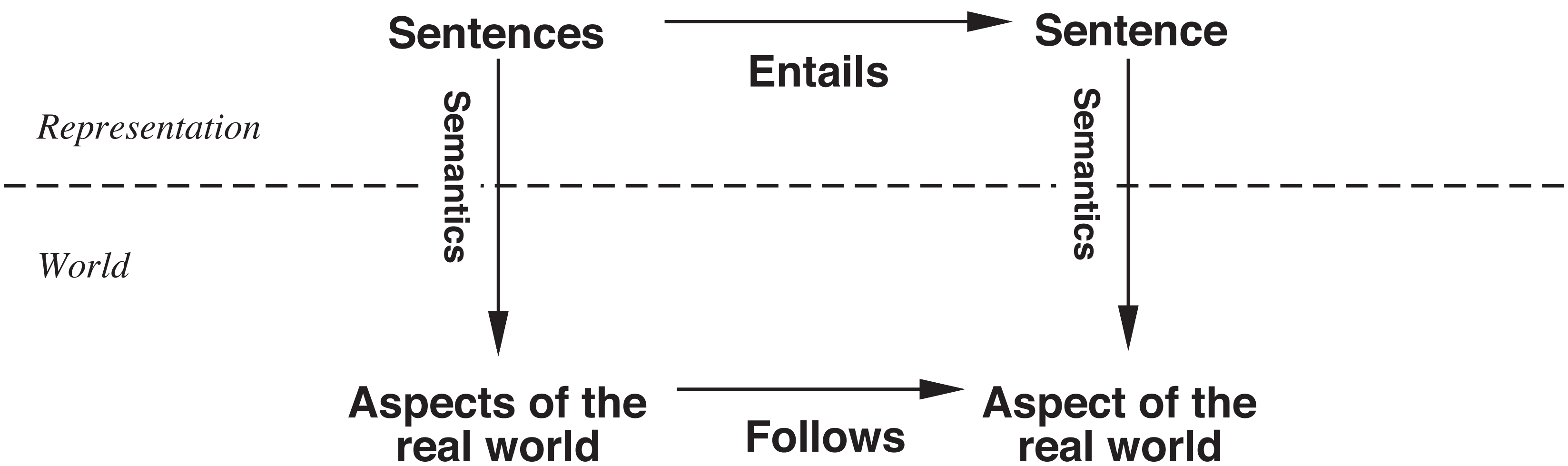
W_1	W_2	W_3	W_4
Giants won Reds won	Giants won	Reds won	

- What worlds are included in each of these:
  - $M(\textit{Giants won}) = \{W_1, W_2\}$
  - $M(\textit{Reds won}) = \{ W_1, W_3\}$
  - $M(\textit{Giants won and Reds won}) = \{W_1\}$
  - $M(\textit{Giants won or Reds won}) = \{W_1, W_2, W_3\}$
- $\mathrm{KB} \vDash \alpha$ iff $M(\mathrm{KB}) \subseteq M(\alpha)$
  - $M(\textit{Giants won or Reds won}) \vDash M(\textit{Giants won})$??
    iff $\{W_1, W_2, W_3\} \subseteq \{W_1, W_2\} \rightarrow$ false!

Inference

- KB $\vdash_i \alpha$ = sentence α can be derived from KB using algorithmic procedure i
- **Soundness**: i is sound if whenever KB $\vdash_i \alpha$, it is also true that KB $\models \alpha$
- **Completeness**: i is complete if whenever KB $\models \alpha$, it is also true that KB $\vdash_i \alpha$
- Preview—later we will define a logic (first-order logic) that is expressive enough to say many things of interest and for which there is a sound and complete inference procedure
  - i.e., the procedure will answer any question whose answer follows from what is known in the KB

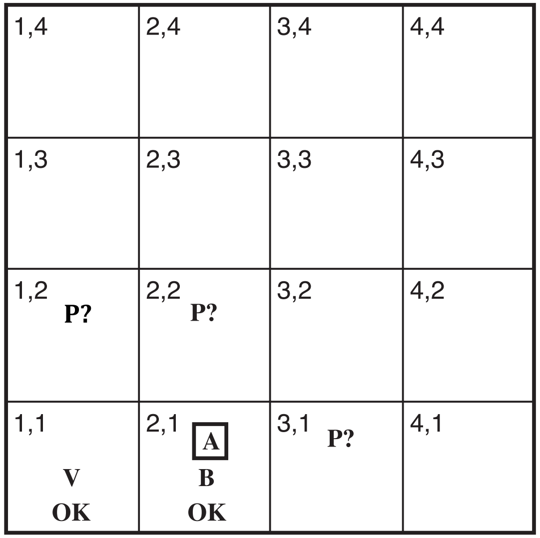
Proof theory and model theory for logic



- Syntax of logic corresponds to aspects of the **real world**
  - Agent uses sensors to relate KB to real environment
  - Learning helps agent construct general rules for inference and understanding

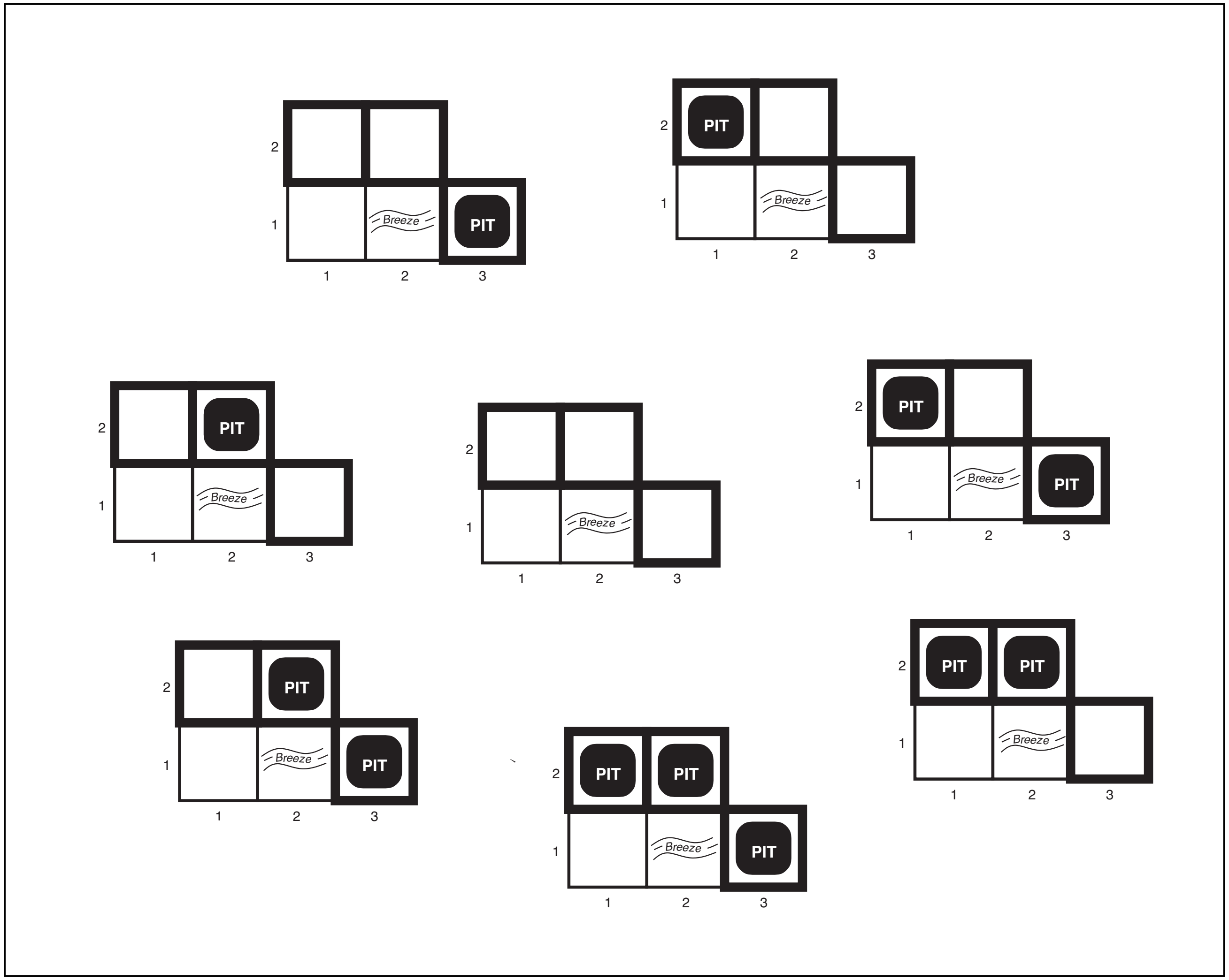
Computing entailment in the wumpus world

- Situation after detecting nothing at [1,1], moving right, and detecting a breeze at [2,1]

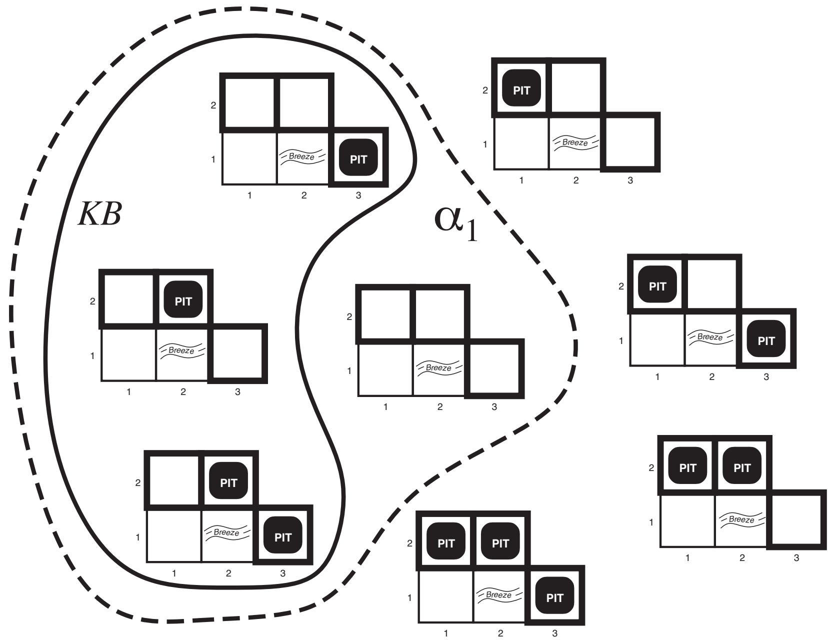


- What can we say about the location of pits in adjacent squares?
  - 3 boolean choices (pit or no pit)→ $2^3 = 8$ possible models for the KB regarding pits

Wumpus models

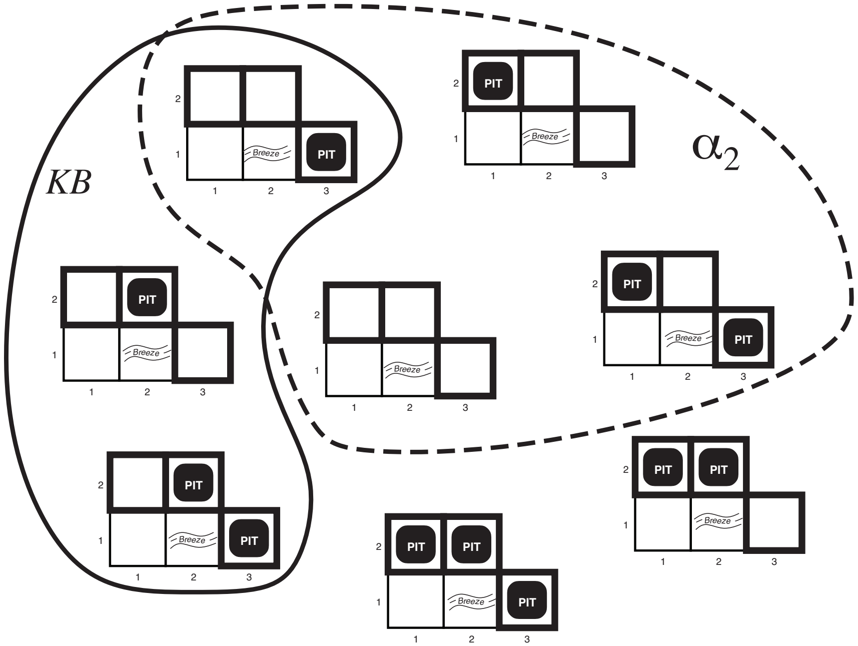


Wumpus models



- KB = wumpus world rules + observations
- α_1 = [1,2] is safe
- KB ⊨ α_1 proved by **model checking**

Wumpus models



- KB = wumpus world rules + observations
- α_2 = [2,2] is safe
- KB $\not\models$ α_2

Propositional logic—syntax

- Propositional logic is the simplest logic
  - Illustrates automated reasoning algorithms
- Allowable sentences
  - Proposition symbols P_1, P_2, … are (atomic) sentences
  - Compound/complex sentences
    - If P is a sentence, $\neg P$ is a sentence (**negation**)
    - If P and Q are sentences, $P \land Q$ is a sentence (**conjunction**)
    - If P and Q are sentences, $P \lor Q$ is a sentence (**disjunction**)
    - If P and Q are sentences, $P \Rightarrow$ Q is a sentence (**implication**)
    - If P and Q are sentences, $P \Leftrightarrow Q$ is a sentence (**biconditional**)
- Operator precedence
  - $\neg$, $\land$, $\lor$, $\Rightarrow$, $\Leftrightarrow$

Propositional logic—semantics

- A model assigns *true*/*false* to each proposition symbol
  - E.g., $m_1 = \{P_{1,2} = false, P_{2,2} = false, P_{3,1} = true\}$
  - 8 possible models can be enumerated with these symbols
- Rules for evaluating truth of **any sentence** w.r.t. model m
  - Atomic sentences—truth of every proposition specified directly by m
  - Compound sentences
    - $\neg P$ is true iff P is false
    - $P \wedge Q$ is true iff P is true **and** Q is true
    - $P \vee Q$ is true iff P is true **or** Q is true
    - $P \Rightarrow Q$ is true iff P is false **or** Q is true
      i.e., is false iff P is true **and** Q is false
    - $P \Leftrightarrow Q$ is true iff $P \Rightarrow Q$ is true **and** $Q \Rightarrow P$ is true

Semantic reasoning—truth tables

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*

- What is up with "material implication" (i.e., ⇒)?
  - Does not require any relation of **causation**
  - Any implication is true when the **antecedent** is false
- For a set of proposition symbols P_1, …, P_N, the max number of models is 2^N
- The number of models of a set of sentences is at most $M = 2^{\text{number of proposition symbols}}$ (**It may be less – why???**)

Logical equivalence

- Two sentences are logically equivalent iff true in same models, i.e., $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

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

Wumpus world knowledge base

- Proposition symbols

  $P_{x,y}$ true if pit at $[x,y]$
  $W_{x,y}$ true if wumpus at $[x,y]$
  $B_{x,y}$ true if breeze perceived at $[x,y]$
  $S_{x,y}$ true if stench perceived at $[x,y]$

- Axioms—true in all wumpus worlds

  $\neg P_{1,1}$.
  $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$.
  $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$.
  ⋮

  - Note: in propositional logic we **can't generalize** knowledge about breezes and adjacent squares
    - Why??? No variables!!! (a simplified answer—more on this later…)

- Percepts

  $\neg B_{1,1}$.
  $B_{2,1}$.

Inference in propositional logic

- Goal is to determine if KB ⊨ α
- First approach—**model checking**
  - **Enumerate** all possible models and check that α is true in all models where KB is true
  - Sound and complete
  - Exponential number of models! Running time $O(2^n)$
- Next approach—**theorem proving**
  - Apply rules of inference to KB to construct **proof** of α
  - Do not have to consult the models directly—can be more efficient
- **Satisfiability**—sentence is true in *some* model
- **Validity**—sentence true in all models (tautology)
  - Deduction theorem: For any sentences α and β, $\alpha \vDash \beta$ iff $\alpha \Rightarrow \beta$ is valid

Syntactic reasoning—sound rules of inference

Name	Premise(s)	Derived Sentence
Modus Ponens	$\alpha, \alpha \Rightarrow \beta$	β
And Introduction	α, β	$\alpha \wedge \beta$
And-Elimination	$\alpha \wedge \beta$	α
Double Negation	$\neg\neg\alpha$	α
Unit Resolution	$\alpha \vee \beta, \neg\beta$	α
Resolution	$\alpha \vee \beta, \neg\beta \vee \Upsilon$	$\alpha \vee \Upsilon$

- When we put the KB into **clause form**, then resolution alone is both sound and complete inference algorithm!

Clauses, inference, and resolution

- **Literal** is an "atomic sentence" (i.e., P, Q, R) or the negation of an atom (i.e., $\neg P$)
- **Clause** is a **disjunction** of literals (i.e., $P \vee \neg Q \vee R$)
- KB is in **Conjunctive Normal Form (CNF)** if represented as a conjunction of disjunctions of literals
  - A set of clauses (AND is implicit) representing the agent's knowledge
- With KB in CNF, resolution is **sound and complete** inference procedure in a single rule!!
- **Theorem**: any set of logic sentences can be transformed into CNF (conjunctive normal form)

Disjunctive and implicative form of clauses

- $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n \vee \boldsymbol{Q \vee R}$—disjunctive form
- Same as: $P_1 \wedge P_2 \wedge \ldots \wedge P_n \Rightarrow \mathbf{Q \vee R}$—implicative form

- What about $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n$?
- Implicative form is: $P_1 \wedge P_2 \wedge \ldots \wedge P_n \Rightarrow \{\,\}$ ($\{\,\} \equiv \mathit{False}$)

- What about Q (single positive literal)?
- Implicative form is: $\Rightarrow$ Q (or just Q)

- Any logical sentence can be converted to (one or **more**) clauses!!

Rules for converting any sentence to CNF

- CNF procedure using **logical equivalence** rules
  1. Replace $P \Leftrightarrow Q$ with $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$ ($\Leftrightarrow$ elimination)
  2. Replace $P \Rightarrow Q$ with $\neg P \vee Q$ ($\Rightarrow$ elimination)
  3. Replace $\neg(\neg P)$ with P (double $\neg$ elim.)
  4. Replace $\neg(P \vee Q)$ with $\neg P \wedge \neg Q$ (De Morgan)
  5. Replace $\neg(P \wedge Q)$ with $\neg \mathrm{P} \vee \neg Q$ (De Morgan)
  6. Apply distributive rule replacing:
     $(P \wedge Q) \vee R$ with $(P \vee R) \wedge (Q \vee R)$ (distributivity)

CNF conversion example

$P \vee Q \Rightarrow R \wedge S$

$\neg(P \vee Q) \vee (R \wedge S)$ (2)

$(\neg P \wedge \neg Q) \vee (R \wedge S)$ (4)

$((\neg P \wedge \neg Q) \vee R) \wedge ((\neg P \wedge \neg Q) \vee S)$ (6)

Clause DB:

$(\neg P \vee R) \wedge (\neg Q \vee R) \wedge (\neg P \vee S) \wedge (\neg Q \vee S)$ (6,6)

- **Convert back to implicative form for intuition ??**

Applying resolution to clauses

- C1. $\neg A \vee \neg B \vee \neg C \vee D \vee E \vee \boldsymbol{F}$
- C2. $\neg P \vee \neg Q \vee \boldsymbol{\neg F} \vee R \vee S$
- These two clauses "**RESOLVE**" because they contain **complementary** literals
- The **resolvent** is:
  $\neg A \vee \neg B \vee \neg C \vee D \vee E \vee \neg P \vee \neg Q \vee R \vee S$
- Intuition: $\neg P \vee Q, P \vee R \Rightarrow Q \vee R$

The resolution rule for propositional logic

$$\frac{[\mathbf{P_1} \vee P_2 \vee \ldots \vee P_n] \wedge [\neg \mathbf{P_1} \vee Q_2 \vee \ldots \vee Q_m]}{P_2 \vee \ldots \vee P_n \vee Q_2 \vee \ldots \vee Q_m}$$

A generalization of modus ponens:

$$\frac{P_1 \wedge [\neg P_1 \vee Q]}{Q}$$

Remember: $[\neg P_1 \vee Q]$ is equivalent to $P_1 \Rightarrow Q$

Example proof by resolution

- Axioms: Qualified ⇒ Hireable
  College-degree ∧ Experience ⇒ Qualified

- Convert axioms to CNF and apply resolution:
  ¬Qualified ∨ Hireable
  ¬College-degree ∨ ¬ Experience ∨ Qualified

  ¬College-degree ∨ ¬ Experience ∨ Hireable

Note: does this mean if a person is hireable they have a college degree and experience? Justify your answer.

Refutation resolution

- Proof by **contradiction**
- Assert the negation of what you want to prove and resolve with current KB to obtain { }
  - Remember { } $\equiv$ *False*
  - Very important when we move to FOL

- Simple example
  $E \Rightarrow A$
  $\neg E \Rightarrow B$
  Prove: $A \vee B$ using refutation resolution

Refutation resolution

- Simple example
  $E \Rightarrow A$
  $\neg E \Rightarrow B$
  Prove: $A \vee B$ using refutation resolution

$(\neg E \vee A) \wedge (E \vee B) \wedge \neg A \wedge \neg B$

$A \wedge B \wedge \neg A \wedge \neg B$

$\{\,\}$

- KB $\wedge \neg(A \vee B)$ is unsatisfiable (reached a contradiction), so KB $\models A \vee B$

Class exercise (R & N Exercise 7.2)

- Given the following, can you prove that the unicorn is mythical? Magical? Horned?

- If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

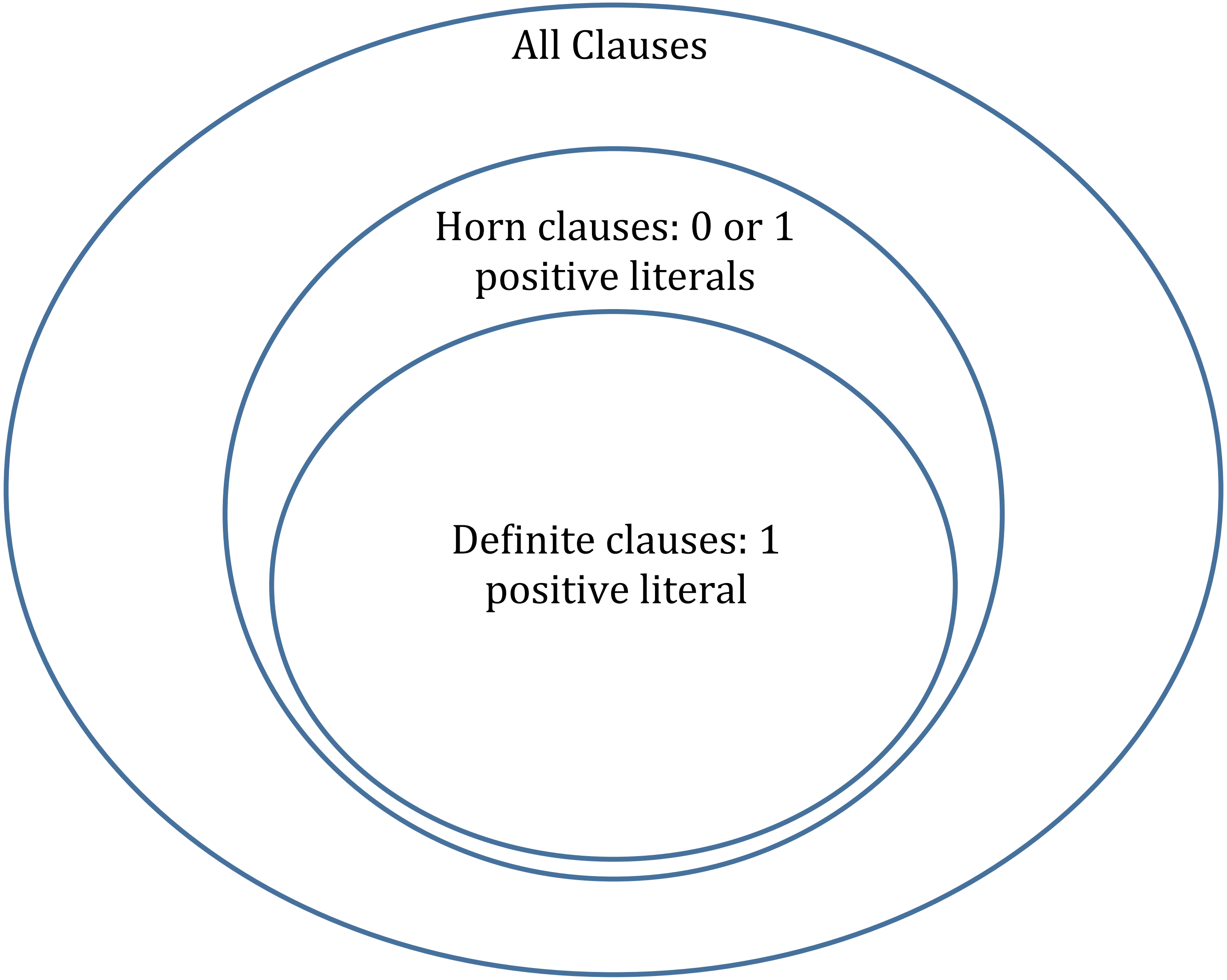
Class exercise (solution)

- Axioms
  1. Mythical ⇒ ¬Mortal
  2. ¬Mythical ⇒ Mortal ∧ Mammal
  3. ¬Mortal ∨ Mammal ⇒ Horned
  4. Horned ⇒ Magical
- Mythical?
  - Unable to determine
- Horned?
  5. Mythical ∨ ¬Mythical (*A* ∨ ¬*A* is a tautology)
  6. ¬Mortal ∨ ¬Mythical (MP 1 and 5)
  7. ¬Mortal ∨ (Mortal ∧ Mammal) (MP 2 and 6)
  8. ¬Mortal ∨ Mammal (And-Elimination)
  9. Horned (MP 3 and 8)
- Magical?
  10. Magical (MP 4 and 9)

Horn clauses

- **Horn clause**—clause with at most one positive literal
  $\neg P_1 \lor \neg P_2 \lor \ldots \lor \neg P_n$
- **Definite clause**—Horn clause with exactly one positive literal
  $\neg P_1 \lor \neg P_2 \lor \ldots \lor \neg P_n \lor R$
- **Goal clause**—Horn clause with no positive literals
  $\neg P_1 \lor \neg P_2 \lor \ldots \lor \neg P_n$
- Closed under resolution (i.e., resolution of Horn clauses will return Horn clause)
- Special properties of KBs with Horn clauses
  1. Definite clauses can be written as implication rules <body> ⇒ <head>
     $(\neg P_1 \lor \neg P_2 \lor \ldots \lor \neg P_n) \Rightarrow R$
  2. Two inference methods that work for Horn clauses
     - **Forward chaining** (data driven)
     - **Backward chaining** (goal driven)
  3. Entailment can be decided in linear time w.r.t. size of KB

Horn clauses and definite clauses



Which of the following are clauses? IF yes, convert to implicative form. Which are Horn clauses? Definite clauses?

1. $A \vee B$
2. $A \wedge B$
3. $\neg A \vee \neg B$
4. $\neg A \wedge \neg B$
5. $\neg A \vee \neg B \vee C \vee D \vee E$
6. $(A \wedge B) \vee C$
7. $\neg(A \wedge \neg B) \vee C$

Which of the following are clauses? IF yes, convert to implicative form. Which are Horn clauses? Definite clauses?

1. $A \vee B$ — **Yes.** $\neg A \Rightarrow B$
2. $A \wedge B$ — **No.**
3. $\neg A \vee \neg B$ — **Yes.** $A \Rightarrow \neg B$**. Horn clause.**
4. $\neg A \wedge \neg B$ — **No.**
5. $\neg A \vee \neg B \vee C \vee D \vee E$ — **Yes.** $A \wedge B \rightarrow C \vee D \vee E$
6. $(A \wedge B) \vee C$ — **No.**
7. $\neg(A \wedge \neg B) \vee C$ — ***No.***

Forward chaining

- Determines if query q is entailed by KB of definite clauses
  - Starts with known facts and derives new knowledge

- **Horn clauses:**
  C1. $\neg P_1 \wedge \neg P_2 \wedge P_4$
  C2. $\neg P_4 \vee P_5$

Rules:
$P_1 \wedge P_2 \Rightarrow P_4$
$P_4 \Rightarrow P_5$

- **Facts:** P_1 , P_2

- Step 1: Percepts P_1 and P_2 resolve with C1 to get P_4 (Add P_4 to KB)
- Step 2: Resolve P_4 with C2 to get P_5
  - This is called **rule chaining**
- Agent can derive conclusions from incoming **percepts**

Backward chaining

- Works backward to determine if the query q is true

- **Horn clauses:**
  C1. $\neg P_1 \wedge \neg P_2 \wedge P_4$
  C2. $\neg P_4 \vee P_5$

Rules:
$P_1 \wedge P_2 \Rightarrow P_4$
$P_4 \Rightarrow P_5$

- **Facts:** P_1 , P_2

Goal: P_5

- Subgoal: prove P_4
  - Sub-sub goal: prove P_2
  - Sub-sub goal: prove P_1
- Very efficient—only touches relevant facts/rules