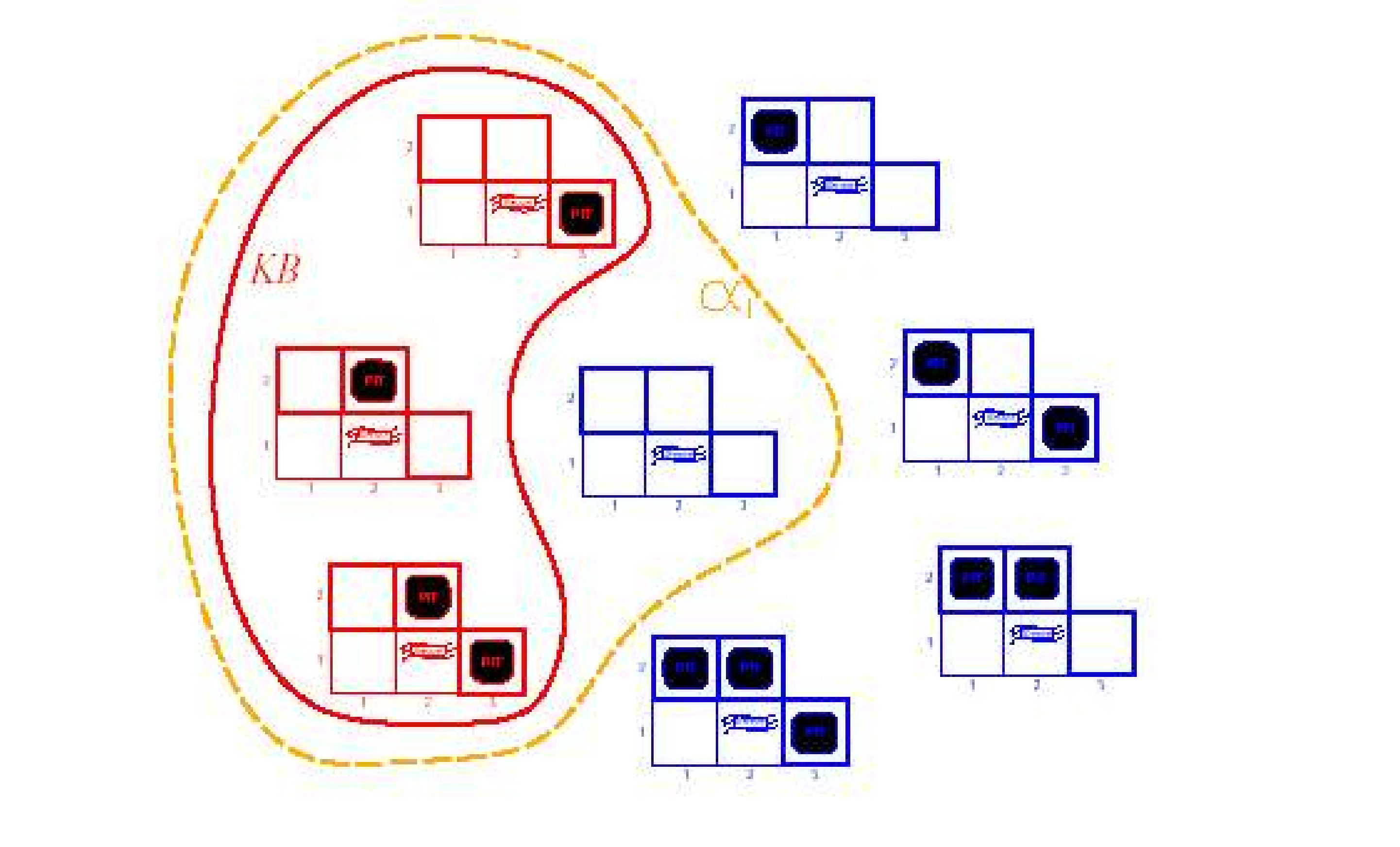
CSC421 Intro to Artificial Intelligence

UNIT 09:
Logical Agents

Wumpus models



KB = wumpus-world rules + observations
a1 = “[1,2]” is safe, KB |= a1 proved by model checking
What about a2 = “[2,2]” ?

Inference

- $KB \models_i a$ sentence a can be derived by procedure i
- Consequences of KB are haystack, a is needle
  - Entailment: needle in haystack
  - Inference: finding it
- Sound: whenever $KB \models_i a$ it is also true that $KB \models a$
- Completeness: i is complete if whenever $KB \models a$ it is also true that $KB \models_i a$

Propositional Logic: Syntax

- Simplest logic – illustrates basic ideas
- The proposition symbols P_1, P_2, etc are sentences
- If S is a sentence then $\neg S$ is a sentence (negation)
- If S_1 and S_2 are sentences then $S_1 \wedge S_2$ is a sentence (conjuction)
- If S_1 and S_2 are sentences then $S_1 \vee S_2$ is a sentence (disjunction)
- If S_1 and S_2 are sentences then $S_1 \Rightarrow S_2$ is a sentence (implication)
- If S_1 and S_2 are sentences then $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Propositional Logic: Semantics

- Each model specifies true/false for each propositional symbol e.g. P_1, csc421isBoring
- Rules for evaluating truth with respect to m:
  - $\neg S$ is true iff S is false
  - $S_1 \wedge S_2$ is true iff S_1 is true and S_2 is true
  - $S_1 \Rightarrow S_2$ is true iff S_1 is false or S_2 is true
  - Etc.
- Simple recursive procedure evaluates arbitrary sentence for a particular model

Truth table for 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*

Wumpus world sentences

- Let P_{ij} be true if there is a pit in [i,j]
- Let B_{ij} be true if there is a pit in [i,j]
- How can you express “Pits cause breeze in adjacent squares” ?

Wumpus world

- Let P_{ij} be true if there is a pit in [i,j]
- Let B_{ij} be true if there is a pit in [i,j]
- How can you express “Pits cause breeze in adjacent squares” ?
  - $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - $B_{2,1} \Leftrightarrow (P_{1,2} \vee P_{2,2} \vee P_{3,1})$
  - etc..

Inference by enumeration

- Depth-first enumeration of all models is sound and complete
- $O(2^N)$ for N symbols
- Straightforward to understand and code
- Works fine for small finite domains

Logical Equivalence

- Two sentences are logically equivalent iff true in same models (a |= b and b |= a)

$$
\begin{array}{rcll}
(\alpha \land \beta) & \equiv & (\beta \land \alpha) & \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv & (\beta \lor \alpha) & \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv & (\alpha \land (\beta \land \gamma)) & \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv & (\alpha \lor (\beta \lor \gamma)) & \text{associativity of } \lor \\
\neg(\neg\alpha) & \equiv & \alpha & \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\beta \Rightarrow \neg\alpha) & \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\alpha \lor \beta) & \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) & \text{biconditional elimination} \\
\neg(\alpha \land \beta) & \equiv & (\neg\alpha \lor \neg\beta) & \text{De Morgan} \\
\neg(\alpha \lor \beta) & \equiv & (\neg\alpha \land \neg\beta) & \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv & ((\alpha \land \beta) \lor (\alpha \land \gamma)) & \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv & ((\alpha \lor \beta) \land (\alpha \lor \gamma)) & \text{distributivity of } \lor \text{ over } \land
\end{array}
$$

Validity and Satisfiability

- A sentence is valid if it is true in all models e.g. A => A,
- A sentence is satisfiable if it is true in some model e.g. $A \vee B$
- A sentence is unsatisfiable if it is true in no models
- Deduction theorem
  - KB |= a iff (KB => a) is valid
- Reduction ad absurdum
  - KB |= a iff $(KB \wedge \neg a)$ is unsatisfiable
- Monotonicity – can't change your mind

Proof methods

- Proof methods are two kinds:
- Application of inference rules
  - Legitimate generation of new sentences from old
  - Proof = sequence of inference rule applications
  - Typically require translation of sentences into normal form
- Model checking
  - Truth table enumeration (exponential)
  - Improved backtracking (DPLL)
  - Heuristic search in model space
    - Sound but not complete
    - Min-conflicts like hill climbing

Forward and Backward Chaining

- Horn Form (restricted)
  - Horn clause =
    - proposition symbol
    - (Conjuction of symbols) ⇒ symbol
- Modus Ponens (for Horn form): complete for Horn KBs

$$\frac{\alpha_1, \ldots, \alpha_n, \qquad \alpha_1 \wedge \cdots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

Can be used for forward chaining, backward chaining
Both run in linear time

Forward Chaining

- Idea:
  - Fire any rule whose premises are satisfied in the KB
  - Add it's conclusion to the KB
  - Repeat until query is found
- Start from what you know, extrapolate until you find what you are seeking
- Do a couple of specific examples on your own
- Sound and complete for Horn form

Backward Chaining

- Idea: work backwards from the query
  - To prove q
    - Check if q is known already
    - Otherwise prove by BC all premises of some rule concluding q
- Avoid loops: check if new subgoal is already on goal stack
- Avoid repeated work: check if new subgoal has been proven true, or has already failed

FC vs BC

- FC is data-driven (automatic, unconscious processing) e.g object recognition, routine decisions
- May do lot's of work that is irrelevant for the goal
- BC is goal-driven (appropriate for problem solving) e.g. Where did I put my keys ?
- Complexity of BC can be much less than linear in size of KB

CNF

- Conjuctive Normal Form (CNF) (universal)
- Conjuction of disjuctions of literals
  - $(A \lor \neg B) \land (B \lor \neg C \lor \neg D)$
- Any KB can be written as a CNF sentence
- It is straightforward to show that any sentence in propositional logic can be converted to CNF
  - Eliminate =>
  - Etc
- Good programming exercise if you like to write parsers

Resolution

- Sound and complete inference for propositional logic

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

Example:

$$\frac{P_{1,3} \vee P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}}$$

Concrete example

- If there is a pit in one of [1,1], [2,2], and [3,1] and there is none in [2,2] then it is either in [1,1] or [3,1]
- KB1: $P_{1,1} \vee P_{2,2} \vee P_{3,1}$
- KB2: $\neg P_{2,2}$
- Resolution of KB1 and KB2 is $P_{1,1} \vee P_{3,1}$

Limitations of Propositional Logic

- Keeping track of location ?
- Single Wumpus ?
- Proliferation of clauses
- Doesn't deal concisely with time, space and universal patterns of relationships
- Need more expressivity – next chapter First-Order Logic

Summary

- Logical agents apply inference to a knowledge base to derive new information and make new decisions
- Basic logic concepts:
  - Syntax, semantics, Entailment, Inference, Soundness, Completeness
- Forward and Backward chaining are linear inference algorithms for Horn clauses
- Resolution is complete for propositional logic
- Propositional logic lacks expressive power