Artificial Intelligence

Prop. Logic

Vibhav Gogate

The University of Texas at Dallas

Some material courtesy of Rina Dechter, Alex Ihler and Stuart Russell, Luke Zettlemoyer, Dan Weld
Outline

• Representing knowledge using logic
  – Agent that reason logically
  – A knowledge based agent

• Representing and reasoning with logic
  – Propositional logic
    • Syntax
    • Semantic
    • validity and models
    • Rules of inference for propositional logic
    • Resolution
    • Complexity of propositional inference.

• Reading: Russel and Norvig, Chapter 7
Knowledge bases

- Knowledge base = set of sentences in a formal language
- Declarative approach to building an agent (or other system):
  - Tell it what it needs to know
- Then it can ask itself what to do - answers should follow from the KB
- Agents can be viewed at the knowledge level
  - i.e., what they know, regardless of how implemented
- Or at the implementation level
  - i.e., data structures in KB and algorithms that manipulate them
Knowledge representation

Defined by syntax and semantics

\[ \Delta \vdash \alpha \]

Assertions (knowledge base) 

\[ \Delta \models \alpha \]

Facts 

Imply 

Conclusions 

Inference 

Computer

Semantics

Real-World

Reasoning: in the syntactic level
Example:

\[ x > y, y > z \models x > z \]
The party example

• If Alex goes, then Beki goes: $A \rightarrow B$
• If Chris goes, then Alex goes: $C \rightarrow A$
• Beki does not go: not $B$
• Chris goes: $C$
• Query: Is it possible to satisfy all these conditions?

• Should I go to the party?
Example of languages

• Programming languages:
  – Formal languages, not ambiguous, but cannot express partial information. Not expressive enough.

• Natural languages:
  – Very expressive but ambiguous: ex: small dogs and cats.

• Good representation language:
  – Both formal and can express partial information, can accommodate inference

• Main approach used in AI: Logic-based languages.
Wumpus World test-bed

• Performance measure
  – gold +1000, death -1000
  – -1 per step, -10 for using the arrow

• Environment
  – Squares adjacent to wumpus are smelly
  – Squares adjacent to 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 in same square
  – Releasing drops the gold in same square

• Sensors: Stench, Breeze, Glitter, Bump, Scream
• Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot
Wumpus world characterization

- Fully Observable? No – only local perception
- Deterministic? Yes – outcomes exactly specified
- Episodic? No – sequential at the level of actions
- Static? Yes – Wumpus and Pits do not move
- Discrete? Yes
- Single-agent? Yes – Wumpus is essentially a natural feature
Aside: NetHack

- WoW of my youth
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Exploring a wumpus world
Propositional Logic
Knowledge Representation and Reasoning System

Big Data → Program or Model → Query?

Learning → Inference
Logic

- **Logics** are formal languages for representing information such that conclusions can be drawn
- **Syntax** defines the sentences in the language
- **Semantics** define the “meaning” of sentences; i.e., define **truth** of a sentence in a world

E.g., the language of arithmetic

- $x + 2 \geq y$ is a sentence; $x^2 + y >$ is not a sentence
- $x + 2 \geq y$ is true iff
  - the number $x + 2$ is no less than the number $y$
- $x + 2 \geq y$ is true in a world where $x = 7, y = 1$
- $x + 2 \geq y$ is false in a world where $x = 0, y = 6$
Entailment and Models

\[ KB \models \alpha \]

- Knowledge base \( KB \) entails sentence \( \alpha \) if and only if \( \alpha \) is true in all worlds where \( KB \) is true.
  
  E.g., the KB containing “the Giants won” and “the Reds won” entails “Either the Giants won or the Reds won”.

- We say \( m \) is a model of a sentence \( \alpha \) if \( \alpha \) is true in \( m \).
  
  \( M(\alpha) \) is the set of all models of \( \alpha \).
  
  Then \( KB \models \alpha \) if and only if \( M(KB) \subseteq M(\alpha) \).
Inference: Deriving Conclusions from a KB

- $KB \vdash_i \alpha = \text{sentence } \alpha \text{ can be derived from } KB \text{ by procedure } i$

- **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**: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure. That is, the procedure will answer any question whose answer follows from what is known by the $KB$. 
Propositional logic is the simplest logic—illustrates basic ideas
The proposition symbols $P_1$, $P_2$ etc are sentences
If $S$ is a sentence, $\neg S$ is a sentence
If $S_1$ and $S_2$ is a sentence, $S_1 \land S_2$ is a sentence
If $S_1$ and $S_2$ is a sentence, $S_1 \lor S_2$ is a sentence
If $S_1$ and $S_2$ is a sentence, $S_1 \Rightarrow S_2$ is a sentence
If $S_1$ and $S_2$ is a sentence, $S_1 \Leftrightarrow S_2$ is a sentence
Propositional logic: Semantics

Each model specifies true/false for each proposition symbol

E.g. \( A \quad B \quad C \)

\[ \text{True} \quad \text{True} \quad \text{False} \]

Rules for evaluating truth with respect to a model \( m \):

\[ \neg S \quad \text{is true iff} \quad S \quad \text{is false} \]

\[ S_1 \land S_2 \quad \text{is true iff} \quad S_1 \quad \text{is true and} \quad S_2 \quad \text{is true} \]

\[ S_1 \lor S_2 \quad \text{is true iff} \quad S_1 \quad \text{is true or} \quad S_2 \quad \text{is true} \]

\[ S_1 \Rightarrow S_2 \quad \text{is true iff} \quad S_1 \quad \text{is false or} \quad S_2 \quad \text{is true} \]

i.e., is false iff \( S_1 \) is true and \( S_2 \) is false

\[ S_1 \Leftrightarrow S_2 \quad \text{is true iff} \quad S_1 \Rightarrow S_2 \quad \text{is true and} \quad S_2 \Rightarrow S_1 \quad \text{is true} \]
Propositional inference: Enumeration method

Let \( \alpha = A \lor B \) and \( KB = (A \lor C) \land (B \lor \neg C) \)
Is it the case that \( KB \models \alpha \)?

Check all possible models—\( \alpha \) must be true wherever \( KB \) is true

Is it sound and complete?

Also called model checking
Propositional Theorem Proving

Apply rules of inference directly to sentences to construct a proof of the desired sentence without consulting models.

If the number of models is large and the proof is small, we have a winner!!

Important concepts for Theorem proving:
- Logical equivalence
- Validity
- Satisfiability
Logical Equivalences

\[(\alpha \land \beta) \equiv (\beta \land \alpha) \quad \text{commutativity of } \land\]
\[(\alpha \lor \beta) \equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor\]
\[((\alpha \land \beta) \land \gamma) \equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land\]
\[((\alpha \lor \beta) \lor \gamma) \equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor\]
\[\neg(\neg \alpha) \equiv \alpha \quad \text{double-negation elimination}\]
\[(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha) \quad \text{contraposition}\]
\[(\alpha \Rightarrow \beta) \equiv (\neg \alpha \lor \beta) \quad \text{implication elimination}\]
\[(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination}\]
\[\neg(\alpha \land \beta) \equiv (\neg \alpha \lor \neg \beta) \quad \text{de Morgan}\]
\[\neg(\alpha \lor \beta) \equiv (\neg \alpha \land \neg \beta) \quad \text{de Morgan}\]
\[(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor\]
\[(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land\]
Validity and Satisfiability

- A sentence is valid if it is true in all models
e.g., \( A \lor \neg A, \quad A \Rightarrow A, \quad (A \land (A \Rightarrow B)) \Rightarrow B \)

- Validity is connected to inference via the Deduction Theorem:
  \( KB \models \alpha \) if and only if \( (KB \Rightarrow \alpha) \) is valid

- A sentence is satisfiable if it is true in some model
e.g., \( A \lor B, \quad C \)

- A sentence is unsatisfiable if it is true in no models
e.g., \( A \land \neg A \)

- Satisfiability is connected to inference via the following:
  \( KB \models \alpha \) if and only if \( (KB \land \neg \alpha) \) is unsatisfiable

Fun facts:

- \( \alpha \) is valid iff \( \neg \alpha \) is unsatisfiable
- \( \alpha \) is satisfiable iff \( \neg \alpha \) is not valid.
Normal forms

Other approaches to inference use syntactic operations on sentences, often expressed in standardized forms:

- **Conjunctive Normal Form** (CNF—universal)
  
  \[
  \text{conjunction of disjunctions of literals} \\
  \text{clauses}
  \]

  E.g., \((A \lor \neg B) \land (B \lor \neg C \lor \neg D)\)

- **Disjunctive Normal Form** (DNF—universal)
  
  \[
  \text{disjunction of conjunctions of literals} \\
  \text{terms}
  \]

  E.g., \((A \land B) \lor (A \land \neg C) \lor (A \land \neg D) \lor (\neg B \land \neg C) \lor (\neg B \land \neg D)\)

- **Horn Form** (restricted)
  
  \[
  \text{conjunction of Horn clauses} \text{ (clauses with} \leq 1 \text{ positive literal)}
  \]

  E.g., \((A \lor \neg B) \land (B \lor \neg C \lor \neg D)\)

  Often written as set of implications:

  \[B \Rightarrow A \text{ and } (C \land D) \Rightarrow B\]
Proof methods divide into (roughly) two kinds:

- **Model checking**
  - truth table enumeration (sound and complete for propositional)
  - heuristic search in model space (sound but incomplete) e.g., the GSAT algorithm (Ex. 6.15)

- **Application of inference rules**
  - Legitimate (sound) generation of new sentences from old
  - Proof = a sequence of inference rule applications
  - Can use inference rules as operators in a standard search alg.
Resolution (for CNF): complete for propositional logic

\[
\alpha \lor \beta, \quad \neg \beta \lor \gamma \\
\hline
\alpha \lor \gamma
\]

Modus Ponens (for Horn Form): complete for Horn KBs

\[
\alpha_1, \ldots, \alpha_n, \quad \alpha_1 \land \cdots \land \alpha_n \Rightarrow \beta \\
\hline
\beta
\]

Can be used with forward chaining or backward chaining
Proof by Resolution

Prove $KB \models \alpha$.

Proof by Contradiction: Prove that $KB \land \neg \alpha$ is unsatisfiable.

- Convert the $KB \land \neg \alpha$ to CNF $\Phi$
- Apply Resolution: Each pair of clauses that contains complementary literals is resolved to produce a new clause which is added to the set $\Phi$. The process continues until one of the following happens:
  1. There are no clauses that can be added in which case the $KB$ does not entail $\alpha$
  2. Two clauses resolve to yield an empty clause in which case the $KB$ entails $\alpha$. 
Conversion to CNF

Method

1. Eliminate $\iff$, replacing $\alpha \iff \beta$ with two formulas $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \alpha$
2. Eliminate $\Rightarrow$ replacing $\alpha \Rightarrow \beta$ by $\neg \alpha \lor \beta$
3. Move $\neg$ inwards.
   1. $\neg(\neg \alpha) = \alpha$
   2. $\neg(\alpha \land \beta) = \neg \alpha \lor \neg \beta$
   3. $\neg(\alpha \lor \beta) = \neg \alpha \land \neg \beta$
4. Distribute $\lor$ over $\land$ where possible
   $\alpha \lor (\beta \land \gamma) = (\alpha \lor \beta) \land (\alpha \lor \gamma)$

Example

- Convert the following KB to CNF:
  $A \iff (B \lor E)$, $E \Rightarrow D$, $C \land F \Rightarrow \neg B$, $E \Rightarrow B$, $B \Rightarrow F$, $B \Rightarrow C$
- Prove by Resolution that $\neg A \land \neg B$ is entailed by the KB.
Given: A KB containing Horn Clauses.
Algorithm: Forward Chaining

- If all premises (left hand side) of an implication are known then its conclusion is added to the set of known facts
- The process continues until the query q is added or until no further inferences can be made.
Proof by Forward and Backward Chaining: Example

AND-OR graphs:
- Multiple links joined by arc indicate a conjunction – every link must be proved.
- Multiple links without an arc indicate a disjunction – any link can be proved.

\[
P \Rightarrow Q \\
L \land M \Rightarrow P \\
B \land L \Rightarrow M \\
A \land P \Rightarrow L \\
A \land B \Rightarrow L \\
A \\
B
\]
Model Checking

1. Convert the $KB \land \neg \alpha$ to CNF
2. The prove that the CNF is unsatisfiable (entailed) or find a solution to the CNF (does not entail)

- Use DPLL + Advanced SAT schemes (Backtracking search).
- Use Walksat (Local search with random restarts and random walks)
Summary

Logical agents apply inference to a knowledge base to derive new information and make decisions.

Basic concepts of logic:
- **syntax**: formal structure of sentences
- **semantics**: truth of sentences wrt models
- **entailment**: necessary truth of one sentence given another
- **inference**: deriving sentences from other sentences
- **soundness**: derivations produce only entailed sentences
- **completeness**: derivations can produce all entailed sentences

Propositional logic suffices for some of these tasks.

Truth table method is sound and complete for propositional logic.

Resolution is sound and complete.