Logic Intro

CPS 170
Ron Parr

Historical Perspective I

• Logic was one of the classical foundations of AI
• Dream: A Knowledge-Based agent
  – Tell the agent facts
  – Agent uses rules of inference to deduce consequences
  – Example: prolog
• Distinction between data and program
• Embodied in field of “Expert Systems”
Example: Minesweeper

- How do you play minesweeper?
- How would you program a machine to do it?
  - Hacking
  - Search
  - Logic
- Logic approach
  - Tell the system of rules of minesweeper
  - System uses logic to make the best moves

What is logic, really?

- Syntax: Rules for constructing valid sentences
- Semantics: Relate syntax to the real world
Entailment

• Aim: Rule for generating (or testing) new sentences that are necessarily true

• The truth of sentence may depend upon the interpretation of the sentence

Interpretations

• An interpretation is a way of matching up objects in the universe with symbols in a sentence (or database).

• A sentence may be true in one interpretation, but false in another

• A necessarily true sentence is true in all interpretations (perhaps given some premises in our KB)
Examples

- Premises (facts in our database):
  - (X or Y)
  - Not X
  - Conclude: Y is necessarily true

- Premises
  - If P then Q
  - Q
  - Conclude: P is not necessarily true
    (though might be true in some interpretations)

Soundness & Completeness

- A (set of) rule(s) of inference is sound if it generates only sentences that are entailed by the knowledge base, i.e., only necessary truths

- A (set of) rule(s) of inference is complete if it can generate all necessary truths

- Can we have one w/o the other?
Historical Perspective II

• Things that are not true necessarily but still true are sometimes said to be “contingent,” “accidental,” or “synthetic,” truths.

• A deep understanding of this distinction evolved through thousands of years of philosophy and mathematics.

• Arguably one of the most important intellectual accomplishments of mankind
  – Basis of mathematic proofs
  – Provides a rigorous procedure for verifying statements

Relation to SAT

• When we want to know if a sentence is satisfiable, what does this mean?

• What about #SAT?

• Why do we care?
Propositional Logic

- Propositional logic is the simplest logic
- All sentences are composed of
  - Atoms
  - Negation
  - Disjunction, conjunction (or, and)
  - Conditional, biconditionals
- Atoms can map to any *proposition* about the universe (depending upon the interpretation)

Checking Validity

- Classic method for checking validity: *truth table*
- Enumerate all possible values (t/f) of atomic elements of a sentence

\[
(P \lor H) \\
\neg H \\
\overline{P}
\]

- Enumerate all 4 (or more) combinations
Inference Rules

• Inference rules are (typically) sound methods of generating new sentences given a set of previous sentences.

• Inference rules save us the trouble of generating truth tables all of the time.

Inference Rules I

• Modus Ponens

\[
\alpha \Rightarrow \beta, \alpha \\
\alpha \\
\beta
\]

• And-Elimination

\[
\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n \\
\alpha_i
\]
### Inference Rules II

- **And-Introduction**

\[
\alpha_1, \alpha_2, \ldots, \alpha_n \quad \frac{\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n}{\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n}
\]

- **Or-Introduction**

\[
\alpha_i \quad \frac{\alpha_i}{\alpha_1 \lor \alpha_2 \lor \ldots \lor \alpha_n}
\]

### Inference Rules III

- **Double Negation Elimination**

\[
\neg \neg \alpha \quad \frac{\neg \neg \alpha}{\alpha}
\]

- **Unit Resolution**

\[
\alpha \lor \beta, \neg \beta \quad \frac{\alpha \lor \beta, \neg \beta}{\alpha}
\]
Resolution

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

Resolution is perhaps the most important inference rule!

Why? Resolution is both sound and complete!

Complexity of Inference

• What is the complexity of exhaustively verifying the validity of a sentence with \( n \) literals (variables)?
  \[ 2^n \]

• Special Case: Horn Logic
  – Horn clauses are disjunctions with at most one positive literal
  – Equivalent to \( P_1 \land P_2 \land \ldots \land P_n \Rightarrow Q \)
Remember De Morgan’s Law?

- \( \neg(P \land Q) = (\neg P) \lor (\neg Q) \)
- \( \neg(P \lor Q) = (\neg P) \land (\neg Q) \)

- Surprisingly, no relationship to Captain Morgan

Implications and Horn Clauses

- If \( P \) then \( Q \)
  - Same as: \( (\neg (P \land (\neg Q))) \)
  - Same as: \( (\neg P) \lor Q \)
  - ...and this is horn!

- If \( (P_1 \land P_2 \land ... \land P_n) \) then \( Q \)
  - Same as: \( (\neg ((P_1 \land P_2 \land ... \land P_n) \land (\neg Q))) \)
  - Same as: \( (\neg (P_1 \land P_2 \land ... \land P_n)) \lor Q \)
  - Same as: \( ((\neg P_1) \lor (\neg P_2) \lor ... \lor (\neg P_n)) \lor Q \)
  - ...and this is horn!
Horn Clause Inference

• Horn clause inference is polynomial – Why?
  – Every sentence establishes exactly one new fact
  – Can add every possible new fact implied by our KB in n passes over our database

• What types of things are easy to represent with horn clauses?
  – Diagnostic rules
  – “Expert Systems”

Shortcomings of Horn Clauses

• Suppose you want to say, “If you have a runny nose and fever, then you have a cold or the flu.”
• If (runny_nose and fever) then (cold or flu)
• But this isn’t a horn clause:
  (not runny_nose) or (not fever) or (cold) or (flu)
• Does adding two separate horn clauses work?
  – (not runny_nose) or (not fever) or (cold)
  – (not runny_nose) or (not fever) or (flu)
Propositional Logic Conclusion

- Logic gives formal rules for reasoning
- Necessarily true = true in all interpretations
- Contrast with CSPs: Satisfiable = true in some, but not necessarily all interpretations
- Sound inference rules generate only necessary truths
- Resolution is a sound and complete inference rule
- Inference with a horn KB is poly time