CS 7180: Behavioral Modeling and Decision-making in AI

Review of Propositional Logic

Prof. Amy Sliva
September 7, 2012
Outline

• General properties of logics
  • Syntax, semantics, entailment, inference, and proofs
• Propositional logic (review)
• Problems with propositional logic
Warning!

“Logic, like whiskey, loses its beneficial effect when taken in too large quantities.”

—Lord Dunsany
What is logic?

- **Logics** are formal languages for representing information such that conclusions can be drawn
  - Expressions in logic are called **sentences**
  - **Declarative** knowledge representation—the “what”, not the “how”

- There are many variations—horn logic, higher-order logic, temporal logic, probabilistic logic, etc...

- **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
Semantic interpretations and entailment

• Semantics determines the **interpretation** of a sentence

• **Truth-functional** semantics
  • Given truth values of all symbols in a sentence, it can be “evaluated” to determine its **truth value** (True or False)

• **Models**—possible worlds
  • Assignments of true or false to every logic symbol
  • We say \( m \) is a model of a sentence \( \alpha \) if \( \alpha \) is true in \( m \)
  • \( M(\alpha) \) denotes **set of all models** of \( \alpha \)—all \( \alpha \) worlds where is true

• **Entailment** means that a sentence follows logically from another

• \( \beta \) **entails** \( \alpha \) (\( \beta \models \alpha \)) iff \( \alpha \) is true in **all** worlds where \( \beta \) is true
  • All models of \( \beta \) are also models of \( \alpha \), i.e., \( M(\beta) \subseteq M(\alpha) \)
Example: semantics of $M(\alpha)$

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

<table>
<thead>
<tr>
<th>$W_1$</th>
<th>$W_2$</th>
<th>$W_3$</th>
<th>$W_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obama speech</td>
<td>Obama speech</td>
<td>Romney speech</td>
<td></td>
</tr>
<tr>
<td>Romney speech</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- What worlds are included in each of these:
  - $M(\text{Obama speech}) = \{W_1, W_2\}$
  - $M(\text{Romney speech}) = \{W_1, W_3\}$
  - $M(\text{Obama speech and Romney speech}) = \{W_1\}$
  - $M(\text{Obama speech or Romney speech}) = \{W_1, W_2, W_3\}$
  - $M(\text{Obama speech or Romney speech}) \models M(\text{Obama speech})$ \iff \{W_1, W_2, W_3\} $\subseteq \{W_1, W_2\}$ $\rightarrow$ false!
Tautologies, contradictions, and satisfiability

- **Valid sentence** or **tautology**
  - Sentence True under all interpretations (in all possible worlds)
  - Example: “It is raining or it is not raining”

- **Inconsistent sentence** or **contradiction**
  - Sentence that is False under all interpretations (in every world)
  - Example: “It is raining and it is not raining.”

- **Satisfiability**—sentence is true in some model
Logical inference

• Logical **inference** creates new sentences that logically follow from a set of sentences \(( \beta )\)

• \( \beta \vdash _i \alpha \) = sentence \( \alpha \) can be derived from \( \beta \) using algorithmic procedure \( i \)

• **Soundness**: \( i \) is sound if whenever \( \beta \vdash _i \alpha \), it is also true that \( \beta \models \alpha \)
  
  • Creates no contradictions

• **Completeness**: \( i \) is complete if whenever \( \beta \models \alpha \), it is also true that \( \beta \vdash _i \alpha \)
  
  • Produces every expression that logically follows (is entailed by) \( \beta \)

• We will look at two specific logics with their own syntax, semantics, and inference procedures
Propositional logic—syntax

• Propositional logic is the simplest logic
• Proposition symbols $P, Q, \ldots$ are atomic sentences
• Wrapping parentheses: ( ... )
• Compound sentences constructed using connectives
  • Operator precedence: $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$
• A sentence is defined as follows:
  • A symbol (atomic sentence) is a sentence
  • If $S$ is a sentence, $\neg S$ is a sentence (negation)
  • If $S$ and $T$ are sentences, $S \land T$ is a sentence (conjunction)
  • If $S$ and $T$ are sentences, $S \lor T$ is a sentence (disjunction)
  • If $S$ and $T$ are sentences, $S \Rightarrow T$ is a sentence (implication)
  • If $S$ and $T$ are sentences, $S \Leftrightarrow T$ is a sentence (biconditional)

• Literal is an atomic sentence or negated atom ($P, \neg P$)
• Sentence formed by finite application of the rules
Propositional logic—semantics

• User defines propositional symbols (i.e., $P$ and $Q$) and semantics of each symbol
  - $P$ means “Obama is POTUS”, $Q$ means “It is sunny”

• A model assigns true/false to each proposition symbol
  - E.g., $m_1 = \{P = false, Q = false, R = true\}$
  - 8 possible models (worlds) 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
    - $P \land Q$ is true iff $P$ is true and $Q$ is true
    - $P \lor Q$ 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

  - $P \Rightarrow Q$ is true and $Q \Rightarrow P$ is true
Semantic reasoning—truth tables

- **Truth tables** are used to determine when a complex sentence is true given the values of the symbols in it.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>¬P</th>
<th>P ∧ Q</th>
<th>P ∨ Q</th>
<th>P ⇒ Q</th>
<th>P ⇔ Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

- What is up with “material implication” (i.e., ⇒)?
  - Does not require any relation of causation
    - "P implies Q" means "it is not the case that P is true and Q false"
  - Any implication is true when the **antecedent** is false
    - ¬P ∨ Q—P is false, or Q is true, or both"
Logical equivalence

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

\[
\begin{align*}
(\alpha \land \beta) & \equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\lnot(\lnot \alpha) & \equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv (\lnot \beta \Rightarrow \lnot \alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv (\lnot \alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \iff \beta) & \equiv (((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\lnot(\alpha \land \beta) & \equiv (\lnot \alpha \lor \lnot \beta) \quad \text{de Morgan} \\
\lnot(\alpha \lor \beta) & \equiv (\lnot \alpha \land \lnot \beta) \quad \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{align*}
\]
Sound rules of inference

- Sound rules of inference for deriving new sentences

<table>
<thead>
<tr>
<th>Name</th>
<th>Premise(s)</th>
<th>Derived Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modus Ponens</td>
<td>( \alpha, \alpha \implies \beta )</td>
<td>( \beta )</td>
</tr>
<tr>
<td>And Introduction</td>
<td>( \alpha, \beta )</td>
<td>( \alpha \land \beta )</td>
</tr>
<tr>
<td>And-Elimination</td>
<td>( \alpha \land \beta )</td>
<td>( \alpha )</td>
</tr>
<tr>
<td>Double Negation</td>
<td>( \neg \neg \alpha )</td>
<td>( \alpha )</td>
</tr>
<tr>
<td>Unit Resolution</td>
<td>( \alpha \lor \beta, \neg \beta )</td>
<td>( \alpha )</td>
</tr>
<tr>
<td>Resolution</td>
<td>( \alpha \lor \beta, \neg \beta \lor \gamma )</td>
<td>( \alpha \lor \gamma )</td>
</tr>
</tbody>
</table>

- **Resolution** alone is a sound and complete inference algorithm!
Proofs in propositional logic

- **Proof** is sequence of sentences where each is a premise or derived from earlier sentences by an inference rule.
- The last sentence is the **theorem** (also called goal or query) that we want to prove.

Example:
1. Qualified $\implies$ Hireable (premise)
2. Qualified (premise)
3. Hireable (modus ponens(1,2))
Resolution is a powerful inference rule

- **Resolution** produces a new clause implied by two clauses containing *complementary literals*
  - **Literal**—atomic sentence (i.e., $P, Q$) or negation of an atom (i.e., $\neg P$)
  - **Clause**—*disjunction* of literals (i.e., $P \lor \neg Q \lor R$)

- Amazingly, this is the only interference rule you need to build a sound and complete proof system
  - Based on proof by contradiction and usually called *resolution refutation*
Application of resolution

- **Conjunctive Normal Form (CNF)**—sentence written as conjunction of disjunctions, i.e., conjunction of clauses
- **Theorem**: any set of sentences can be transformed into CNF
- **Knowledge base** is a set of true propositional sentences, i.e., conjunction of sentences (\(\land\) is implicit)

Example: \(KB = [P \Rightarrow Q, Q \Rightarrow R \land S]\)

- \(KB\) in CNF: \([\neg P \lor Q, \neg Q \lor R, \neg Q \lor S]\)
- Resolve \(\neg P \lor Q\) and \(\neg Q \lor R\) producing: \(\neg P \lor R\) (i.e., \(P \Rightarrow R\))
  - Intuition: \((\neg P \lor Q, \neg Q \lor R) \Rightarrow P \lor R\)
- Resolve \(\neg P \lor Q\) and \(\neg Q \lor S\) producing: \(\neg P \lor S\) (i.e., \(P \Rightarrow S\))
- New \(KB = [\neg P \lor Q, \neg Q \lor R, \neg Q \lor S, \neg P \lor R, \neg P \lor S]\)
The resolution rule for propositional logic

\[ [P_1 \lor P_2 \lor \ldots \lor P_n], [\neg P_1 \lor Q_2 \lor \ldots \lor Q_m] \]

\[ \frac{}{P_2 \lor \ldots \lor P_n \lor Q_2 \lor \ldots \lor Q_m} \]

A generalization of modus ponens:

\[ P_1, \neg P_1 \lor Q \] \quad \text{Remember: } \neg P_1 \lor Q \text{ is equivalent to } P_1 \implies Q

\[ \frac{}{Q} \]
Example proof by resolution

• Premises: Qualified ⇒ Hireable
  College-degree ∧ Experience ⇒ Qualified

• Convert premises to CNF and apply resolution:
  1. ¬Qualified ∨ Hireable  Premise
  2. ¬College-degree ∨ ¬Experience ∨ Qualified  Premise

  ________________________________________________
  3. ¬College-degree ∨ ¬Experience ∨ Hireable  Resolution
     (College-degree ∧ Experience ⇒ Hireable)
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>
     \[ P_1 \land P_2 \land \ldots \land P_n \Rightarrow R \]
  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
Forward chaining

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

- **Horn clauses:**
  - $C_1$: $\neg P_1 \lor \neg P_2 \lor P_4$
  - $C_2$: $\neg P_4 \lor P_5$

- **Rules:**
  - $P_1 \land P_2 \Rightarrow P_4$
  - $P_4 \Rightarrow P_5$

- **Facts:** $P_1$, $P_2$

- Step 1: Facts $P_1$ and $P_2$ resolve with $C_1$ to get $P_4$ (Add $P_4$ to KB)
- Step 2: Resolve $P_4$ with $C_2$ to get $P_5$
  - This is called **rule chaining**
- Derive conclusions from incoming **facts**
Backward chaining

• Works backward to determine if the query $q$ is true

• Horn clauses:
  C1. $\neg P_1 \lor \neg P_2 \land P_4$
  C2. $\neg P_4 \lor P_5$

• Facts: $P_1, P_2$

• Rules:
  $P_1 \land P_2 \Rightarrow P_4$
  $P_4 \Rightarrow P_5$

• 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
Hunt the Wumpus in propositional logic

• Some atomic propositions:
  S12 = There is a stench in cell (1,2)
  B34 = There is a breeze in cell (3,4)
  W22 = Wumpus is in cell (2,2)
  V11 = We’ve visited cell (1,1)
  OK11 = Cell (1,1) is safe.
  ...

• Some rules:
  (R1) \( \neg S11 \Rightarrow W11 \land \neg W12 \land \neg W21 \)
  (R2) \( \neg S21 \Rightarrow W11 \land \neg W21 \land \neg W22 \land \neg W31 \)
  (R3) \( \neg S12 \Rightarrow W11 \land \neg W12 \land \neg W22 \land \neg W13 \)
  (R4) \( S12 \Rightarrow W13 \lor W12 \lor W22 \lor W11 \)
  ...

• The lack of variables requires us to give similar rules for each cell!
Inference and proof in the Wumpus world

- Prove that the Wumpus is in (1,3) using the four rules given
  1. Modus ponens with \( \neg S_{11} \) and \( R_1 \):
     \( \neg W_{11} \land \neg W_{12} \land \neg W_{21} \)
  2. And-Elimination to this, yielding 3 sentences:
     \( \neg W_{11}, \neg W_{12}, \neg W_{21} \)
  3. Modus ponens with \( \neg S_{21} \) and \( R_2 \), then apply And-Elimination:
     \( \neg W_{22}, \neg W_{21}, \neg W_{31} \)
  4. Modus ponens with \( S_{12} \) and \( R_4 \) to obtain:
     \( W_{13} \lor W_{12} \lor W_{22} \lor W_{11} \)
  5. Unit resolution on \( (W_{13} \lor W_{12} \lor W_{22} \lor W_{11}) \) and \( \neg W_{11} \):
     \( W_{13} \lor W_{12} \lor W_{22} \)
  6. Unit Resolution with \( (W_{13} \lor W_{12} \lor W_{22}) \) and \( \neg W_{22} \):
     \( W_{13} \lor W_{12} \)
  7. Apply Unit Resolution with \( (W_{13} \lor W_{12}) \) and \( \neg W_{12} \):
     \( W_{13} \)
QED
Problems with propositional Wumpus hunter

- Not possible to state more **general** rules
  - Need a similar set of rules for each cell
- Dynamic knowledge difficult to represent
  - Standard approach is to index facts with temporal information
    \[ L_{1,1,0} \land up_{0} \Rightarrow L_{2,1,1} \]—one for each location \([x,y]\) at time \(t\)
    \[ L_{1,1,0} \land has-arrow_{0} \land shoot_{0} \Rightarrow L_{1,1,1} \land \neg has-arrow_{1} \]
- The frame problem requires exhaustive representation of effects (and non-effects)
  \[ L_{1,1,0} \land up_{0} \Rightarrow L_{2,1,1} \land \neg L_{1,1,1} \]
  - But it gets even worse!
    \[ L_{1,1,0} \land has-arrow_{0} \land up_{0} \Rightarrow L_{2,1,1} \land \neg L_{1,1,1} \land has-arrow_{1} \]
    \[ L_{1,1,0} \land \neg has-arrow_{0} \land up_{0} \Rightarrow L_{2,1,1} \land \neg L_{1,1,1} \land \neg has-arrow_{1} \]
Pros and cons of propositional logic

- Propositional logic is **declarative**
- PL allows **partial/disjunctive/negated** information
  - Horn clauses are a nice intermediate form
- Propositional logic is **compositional**
  - Meaning of $L_{1,1} \land up$ is derived from the meaning of $L_{1,1}$ and of $up$
- Meaning in PL is **context-independent**
  - Unlike natural language, where meaning depends on context

- Propositional logic has very **limited expressive power**
  - Unlike natural language...
  - Can become impractical even for simple worlds