Reviewing Lecture Material

Propositional Logic

First Order Logic

Summary

Problems
Logic

- Thinking in terms of logical formulas and inference rules, as opposed to state or variable based models.
- "Logic language" to represent and reason with knowledge.
  - Syntax: defines valid formulas.
  - Semantics: specify models (satisfying assignments) for each formula.
  - Inference rules: what does a formula imply?
Reviewing Lecture Material

Propositional Logic
Syntax and Semantics of Propositional Logic

Syntax

- **Propositional symbols**: $A, B, C, \text{Cat}, \ldots$
  - Can be anything
- **Logical connectives**: $\neg, \land, \lor, \rightarrow, \leftrightarrow$
  - Not, And, Or, Implies, Equals
- Build up formulas recursively, can operate on formulas with logical connectives

Semantics

**Definition**
A model in propositional logic is an **assignment** of truth values to propositional symbols.

**Interpretation function**: if model $w$ satisfies formula $f$, then

$$ I(f, w) \in \{0, 1\} $$
• Let $\mathcal{M}(f)$ be the set of models $w$ for which $I(f, w) = 1$.
  • Set of all possible valid assignments. A formula compactly represents a set of models.

• A knowledge base $KB$ is a set of formulas representing their intersection:

$$\mathcal{M}(KB) = \bigcap_{f \in KB} \mathcal{M}(f)$$

• KB specifics constraints on the world, $\mathcal{M}(KB)$ is the set of all worlds satisfying those constraints.

• Remember $\mathcal{M}(f)$ is just a set of models, this is intersection of those sets over a number of $f$.

• Adding more formulas to the knowledge base ...
Models

• Let $M(f)$ be the set of models $w$ for which $I(f, w) = 1$.
  • Set of all possible valid assignments. A formula compactly represents a set of models.

• A knowledge base $KB$ is a set of formulas representing their intersection:
  \[ M(KB) = \bigcap_{f \in KB} M(f) \]

  • KB specifics constraints on the world, $M(KB)$ is the set of all worlds satisfying those constraints.
  • Remember $M(f)$ is just a set of models, this is intersection of those sets over a number of $f$.

• Adding more formulas to the knowledge base ... shrinks the set of models
Adding Formulas

When we add $f$ to KB:

1. **Entailment**: no information was added.
   - $\text{KB} \models f$ iff $\mathcal{M}(\text{KB}) \subseteq \mathcal{M}(f)$

2. **Contradiction**: $f$ contradicts what we know
   - $\text{KB}$ contradicts $f$ iff $\mathcal{M}(\text{KB}) \cap \mathcal{M}(f) = \emptyset$

3. **Contingency**: $f$ adds non-trivial information to $KB$
   - $\emptyset \subsetneq \mathcal{M}(\text{KB}) \cap \mathcal{M}(f) \subsetneq \mathcal{M}(\text{KB})$

KB contradicts $f$ iff KB entails $\neg f$. 
Inference Rules

Inference rules allow us to reason with formulas without ever instantiating models.

Modus Ponens: for any propositional symbols $p$ and $q$:

$$\frac{p, \ p \rightarrow q}{q} \quad \text{which is} \quad \frac{(\text{premises})}{(\text{conclusion})}$$

Inference rule: (syntax, not semantics!) if $f_1, \ldots, f_k, g$ are formulas then:

$$\frac{f_1, \ldots, f_k}{g}$$

KB derives/proves $f$ ($KB \vdash f$) iff $f$ eventually is added to KB through inference rules.
A CNF formula is a conjunction (and) of clauses (or’s):

\[(A \lor B \lor \neg C) \land (\neg B \lor D)\]

Can always convert:

- \(f \leftrightarrow g\) is \((f \rightarrow g) \land (g \rightarrow f)\)
- \(f \rightarrow g\) is \(\neg f \lor g\)
- \(\neg (f \land g)\) is \(\neg f \lor \neg g\)
- \(\neg (f \lor g)\) is \(\neg f \land \neg g\)
- Double negatives cancel.
- Can distribute \(\lor\) over \(\land\): \(f \lor (g \land h)\) is \((f \lor g) \land (f \lor h)\)
Resolution

Remember that entailment $KB \models f$ is the opposite of contradiction $KB \cup \{\neg f\}$ is unsatisfiable.

Resolution-based inference:

- Add $\neg f$ into $KB$
- Convert all formulas into CNF
- Repeatedly apply resolution rule.
- Return entailment iff derive false.

Resolution rule:

$$f_1 \lor \cdots \lor f_n \lor p, \quad \neg p \lor g_1 \lor \cdots \lor g_m$$

$$\quad \frac{f_1 \lor \cdots \lor f_n \lor g_1 \lor \cdots \lor g_m}{f_1 \lor \cdots \lor f_n \lor g_1 \lor \cdots \lor g_m}$$
Reviewing Lecture Material

First Order Logic
Syntax

Uses terms to refer to objects:

- Constant symbols (e.g. arithmetic)
- Variable (e.g. $x$)
- Functions of terms (e.g. $\text{Sum}(3, x)$)

Formulas refer to truth values:

- Atomic formulas (atoms), predicate applied to terms.
  - $\text{Knows}(x, \text{arithmetic})$
- Connectives applied to formulas:
  - $\text{Student}(x) \rightarrow \text{Knows}(x, \text{arithmetic})$
- Quantifiers applied to formulas:
  - $\forall x \text{Student}(x) \rightarrow \text{Knows}(x, \text{arithmetic})$
Quantifiers

- **Universal quantification** (\(\forall\)): 
  - Like conjunction (and): \(\forall x P(x)\) is \(P(A) \land P(B) \land \cdots\)

- **Existential quantification** (\(\exists\)): 
  - Like disjunction (or): \(\exists x P(x)\) is like \(P(A) \lor P(B) \lor \cdots\)

**Properties:**
- \(\neg \forall x P(x)\) is equivalently \(\exists x \neq P(x)\)
- \(\forall x \exists y \text{Knows}(x, y)\) is different from \(\exists y \forall x \text{Knows}(x, y)\).

**Models** in first order logic map constant symbols to objects and predicate symbols to (satisfying) tuples of objects.
Summary

Reviewing Lecture Material

- Propositional Logic
- First Order Logic

Summary

Problems
Things we Skipped Reviewing

- Soundness and completeness
- Propositional Horn clauses (definite + goal clause)
- First order modens ponens
- First order resolution
Problems

Reviewing Lecture Material

- Propositional Logic
- First Order Logic

Summary

Problems
**Definition**

A **definite clause** has the following form:

\[(p_1 \land \cdots \land p_k) \rightarrow q\]

\(p_i\) and \(q\) are propositional symbols. Formula, not an inference rule.

A **Horn clause** is either:

- A definite clause \((p_1 \land \cdots \land p_k) \rightarrow q\)
- A goal clause \((p_1 \land \cdots \land p_k) \rightarrow \text{false}\), equivalently \(\neg(p_1 \land \cdots \land p_k)\).