Entailment

- \( \alpha \) entails \( \beta \) or \( \alpha \models \beta \)
  - \( \beta \) follows logically from \( \alpha \)
  - In every model in which \( \alpha \) is true, \( \beta \) is also true
    - \( M(\alpha) \subseteq M(\beta) \)

Class Overview

- Review from Wednesday
- Inference in propositional logic
- Propositional logic agents
- First-Order Logic (Ch 8)
Propositional Logic Syntax

- \( \text{Sentence} \rightarrow \text{AtomicSentence} \mid \text{ComplexSentence} \)
- \( \text{AtomicSentence} \rightarrow \text{True} \mid \text{False} \mid P \mid Q \mid R \mid \ldots \)
- \( \text{Complex Sentence} \rightarrow (\text{Sentence}) \mid [\text{Sentence}] \)
  - \( \neg \text{Sentence} \mid \text{Sentence} \land \text{Sentence} \)
  - \( \text{Sentence} \lor \text{Sentence} \mid \text{Sentence} \Rightarrow \text{Sentence} \)
  - \( \text{Sentence} \Leftrightarrow \text{Sentence} \)
- Operator precedence: \( \neg, \land, \lor, \Rightarrow, \Leftrightarrow \)

Example statements

- There is no pit in [1, 1]
- A square is breezy iff there is a pit in a neighboring square
- If there is no smell in [1, 1], there can’t be a wumpus in [1, 2]

Model checking

- How does it work?
- What is the running time?
- What is the space required?

Theorem proving [7.5]

- No longer consult models
  - Derive inferences (entailment) directly from KB
- In some ways this mimics algebraic theorem proving
  - Start with the known
  - Apply rules/transformations
  - Reach the desired result (if possible)
Logical equivalence

- Two statements are logically equivalent if they are true in the same set of models
  - $\alpha = \beta$
  - $\alpha = \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

Standard logical equivalences

- $(\alpha \land \beta) = (\beta \land \alpha)$
- $(\alpha \lor \beta) = (\beta \lor \alpha)$
- $((\alpha \land \beta) \land \gamma) = (\alpha \land (\beta \land \gamma))$
- $((\alpha \lor \beta) \lor \gamma) = (\alpha \lor (\beta \lor \gamma))$
- $\neg (\neg \alpha) = \alpha$
Validity

- A sentence is valid if it is true in all models
  - $P \lor \neg P$
  - $Q \Rightarrow Q$

- Valid sentences are tautologies

**Deduction** theorem
- For any sentences $\alpha$ and $\beta$, $\alpha \models \beta$ iff $(\alpha \Rightarrow \beta)$ is valid
- Essence of model checking algorithm

Satisfiability

- A sentence is satisfiable if it is true in some model
  - Abbreviated as SAT
  - Can we find a variable assignment that makes some statement true

Validity and Satisfiability

- $\alpha$ is satisfiable iff $\neg \alpha$ is not valid
- $\alpha \models \beta$ iff $(\alpha \land \neg \beta)$ is unsatisfiable
  - Proof? [Hint: $\alpha \models \beta$ iff $(\alpha \Rightarrow \beta)$ is valid]
- This is the logical basis of proof by contradiction

Validity and Satisfiability (Proof)

- $\alpha$ is satisfiable iff $\neg \alpha$ is not valid
  - if $\alpha$ is unsatisfiable, $\neg \alpha$ is valid
  - if $\neg \alpha$ is unsatisfiable, $\alpha$ is valid
- $\alpha \models \beta$ iff $(\alpha \land \neg \beta)$ is unsatisfiable
  - $\alpha \models \beta$ iff $(\alpha \Rightarrow \beta)$ is valid
  - $\alpha \models \beta$ iff $\neg (\alpha \Rightarrow \beta)$ is unsatisfiable
  - $\alpha \models \beta$ iff $\neg (\neg \alpha \lor \beta)$ is unsatisfiable
  - $\alpha \models \beta$ iff $(\alpha \land \neg \beta)$ is unsatisfiable
Inference & Proofs

- New notation for inference rules
  \[
  \text{given}_1, \quad \text{given}_2 \quad \text{conclusion}
  \]
- We supply the items on the top and conclude what is on the bottom

Modus Ponens

- Latin for mode that affirms
  \[
  \alpha \Rightarrow \beta, \quad \alpha \quad \Rightarrow \beta
  \]

And-Elimination

\[
\frac{\alpha \land \beta}{\alpha}
\]

Biconditional elimination

\[
\frac{\alpha \leftrightarrow \beta}{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}
\]

\[
(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)
\]

\[
\alpha \leftrightarrow \beta
\]
**Book Examples**

- Question 7.4

**Search**

- We can formulate theorem proving as a search problem
  - Initial state: KB
  - Actions: all inference rules that apply (top of rule)
  - Result: inference in bottom of rule added to KB
  - Goal: sentence we want to prove

**Monotonicity**

- The set of entailed sentences can only increase as information is added to the KB
  - if KB |= α then KB ∧ β |= α
  - Adding β to our KB will not decrease what we can entail from the KB

**Inference: sound & complete**

- The previous inference rules were all sound
  - Derive entailed sentences
- Are they complete? No
  - There are some things they can’t derive
    - (Example?)
Unit Resolution

\[ l_1 \lor l_2, \neg l_2 \]
\[ \overset{\text{Resolution}}{\Rightarrow} l_1 \]

- Can be generalized to more clauses (see book)

Resolution

- Generalized resolution can handle more clauses

\[ l_1 \lor l_2, \neg l_2 \lor l_3 \]
\[ \overset{\text{Resolution}}{\Rightarrow} l_1 \lor l_3 \]

- Completely general form in the book

Examples

Conjunctive Normal Form (CNF)

- Resolution only applies to clauses with disjunction (\( \lor \))
  - All propositional logic can be reduce to clauses or conjunctive normal form (CNF)
**Using resolution**

- Proofs using resolution are proofs by contradiction
  - \( \alpha \models \beta \) iff \((\alpha \land \neg \beta)\) is unsatisfiable
- Assume we want to prove \( \alpha \models \beta \)
  - Add \( \neg \beta \) to KB
  - If we can infer \( \text{false} \), we have a contradiction
  - If we can’t, then \( \alpha \not\models \beta \)

**Example**

- \( B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1}) \)
- \( B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1}) \land (P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1} \)
- \((\neg B_{1,1} \lor (P_{1,2} \lor P_{2,1})) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1}) \)
- \((\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1}) \)
  - \((\neg P_{1,2} \lor \neg P_{2,1}) \lor B_{1,1}\)
  - \(((B_{1,1} \lor \neg P_{1,2}) \land (B_{1,1} \lor \neg P_{2,1}))\)
  - \((\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (B_{1,1} \lor \neg P_{1,2}) \land (B_{1,1} \lor \neg P_{2,1})\)

**Special Case: Horn & definite Clauses**

- A Horn clause is a disjunction of literals of which *at most one* is positive
  - \( \neg A \lor \neg B \lor C \)
  - In Definite clause *exactly one* is positive
  - Definite clauses correspond to implications
    - \( A \land B \Rightarrow C \)
  - Modus Ponens is sound and complete with Horn clauses
Building Logic Agents

- Can we now build propositional logic agents?
  - There are a few important details!
- All percepts depend on the current time/location of the agent
  - Frame problem: need to reason about what does/does not change as time goes forward
  - This tremendously complicates writing proper logical descriptions of the world

First-Order Logic: Motivation

- Returning to fred likes bones:
  - Expensive to have to specify if everyone likes bones
  - Works in wumpus world, but can be computationally infeasable
  - Cannot make statements like:
    - “All dogs like bones”

Building Logic Agents

- Can now build an agent
  - Use A* to plan movement
  - Use logical inference to decide where to go
  - Caveat: planning gets more expensive as more time passes, even if the agent just moves around the know part of the state space
  - Harder to build an agent that generates a full plan

First-Order Logic

- Propositional logic only has variables
  - These are true or false
- First-order logic adds objects, functions and relations
- Also adds quantifiers:
  - $\exists$: There exists
  - $\forall$: For all
First-Order Logic Examples

- \textit{Occupation}(p, o); \textit{Boss}(p1, p2); \textit{Customer}(p1, p2)
- \textit{Emily}; \textit{Doctor, Surgeon, Lawyer}

- Emily is either a surgeon or a lawyer.
- All surgeons are doctors.
- Emily has a boss who is a lawyer.
- Every surgeon has a lawyer.

Homework: 8.10