Resolution: Summary

- FOL resolution rule:
  \[
  \frac{l_1 \lor \cdots \lor l_k \lor m_1 \lor \cdots \lor m_n}{(l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \lor \cdots \lor l_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n)\varnothing}
  \]
  where \(\text{Unify}(l_i, \neg m_j) = \varnothing\).

- The two clauses are assumed to be standardized apart so that they share no variables.

- Example:
  \[
  \begin{align*}
  \neg \text{Rich}(x) \lor \text{HasSwissBankAccount}(x) \\
  \text{Rich}(\text{Willard})
  \end{align*}
  \]
  \[
  \frac{\text{HasSwissBankAccount}(\text{Willard})}{
  \text{Rich}(\text{Willard}) \lor \text{HasSwissBankAccount}(\text{Willard})}
  \]

  with \(\varnothing = \{x/\text{Willard}\}\)
Resolution: Conversion to CNF

Everyone who loves all animals is loved by someone:
\[ \forall x \ [ \forall y \ Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y \ Loves(y,x)] \]

1. Eliminate biconditionals and implications
\[ \forall x [\forall y \neg Animal(y) \lor Loves(x,y)] \lor [\exists y \ Loves(y,x)] \]

2. Move \neg inwards: \neg \forall x p \equiv \exists x \neg p, \neg \exists x p \equiv \forall x \neg p
\[ \forall x [\exists y (\neg Animal(y) \lor Loves(x,y))] \lor [\exists y \ Loves(y,x)] \]
\[ \forall x [\exists y Animal(y) \land \neg Loves(x,y)] \lor [\exists y \ Loves(y,x)] \]

Conversion to CNF contd.

3. Standardize variables: Each quantifier uses a different variable
\[ \forall x [\exists y Animal(y) \land \neg Loves(x,y)] \lor [\exists z Loves(z,x)] \]

4. Skolemize: Each existential variable is replaced by a Skolem function of the enclosing universally quantified variables:
\[ \forall x [Animal(F(x)) \land \neg Loves(x,F(x))] \lor Loves(G(x),x) \]

5. Drop universal quantifiers:
\[ [Animal(F(x)) \land \neg Loves(x,F(x))] \lor Loves(G(x),x) \]

6. Distribute \lor over \land to get CNF (clauses connected by \land):
\[ [Animal(F(x)) \lor Loves(G(x),x)] \land [\neg Loves(x,F(x)) \lor Loves(G(x),x)] \]
Example: Nono and West again

- It is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles. All of its missiles were sold to it by Colonel West, who is American.

- Is Col. West a criminal?

- FOL representation:

\[ \forall x \text{ American}(x) \land \text{Weapon}(y) \land \text{Sells}(x,y,z) \land \text{Hostile}(z) \Rightarrow \text{Criminal}(x) \]
\[ \exists x \text{ Owns}(\text{Nono},x) \land \text{Missile}(x) \]
\[ \forall x \text{ Missile}(x) \land \text{Owns}(\text{Nono},x) \Rightarrow \text{Sells}(\text{West},x,\text{Nono}) \]
\[ \forall x \text{ Missile}(x) \Rightarrow \text{Weapon}(x) \]
\[ \forall x \text{ Enemy}(x,\text{America}) \Rightarrow \text{Hostile}(x) \]
\[ \text{American}(\text{West}) \]
\[ \text{Enemy}(\text{Nono},\text{America}) \]

KB in CNF and Resolution

- KB in CNF (variables not standardized):

- American\(^{-}\) (variables not standardized):

\[ \neg \text{American}(x) \lor \neg \text{Weapon}(y) \lor \neg \text{Sells}(x,y,z) \lor \neg \text{Hostile}(z) \lor \neg \text{Criminal}(x) \]
\[ \text{Owns}(\text{Nono,M}_1) \text{ [Skolem constant M}_1 \text{]} \]
\[ \text{Missile}(M_1) \]
\[ \neg \text{Missile}(x) \lor \neg \text{Owns}(\text{Nono},x) \lor \neg \text{Sells}(\text{West},x,\text{Nono}) \]
\[ \neg \text{Missile}(x) \lor \neg \text{Weapon}(x) \]
\[ \neg \text{Enemy}(x,\text{America}) \lor \neg \text{Hostile}(x) \]
\[ \text{American}(\text{West}) \]
\[ \text{Enemy}(\text{Nono, America}) \]

- Resolution: Uses “proof by contradiction”

Show \( KB \models \alpha \) by showing \( KB \land \neg \alpha \) unsatisfiable

- To prove Col. West is a criminal, add \( \neg \text{Criminal}(\text{West}) \) to \( KB \) and derive empty clause
FOL Resolution Example

\[ \neg \text{American}(x) \lor \neg \text{Weapon}(y) \lor \neg \text{Sells}(y,z) \lor \neg \text{Hostile}(z) \lor \text{Criminal}(x) \]

Contradiction!

Therefore,

\[ \text{Criminal}(\text{West}) \]

FOL Resolution Example 2

Given

\[ \forall x \exists y \, \text{Twin}(x) \Rightarrow \text{Twin}(y) \]

Twin(Ashley)

\[ \forall x \, \text{Twin}(x) \Rightarrow \text{Twin}(F(x)) \]

Skolemization

\[ [ \neg T(x), T(F(x))), T(A), \neg T(D)] \]

T(F(A))

T(F(F(A)))

T(F(F(F(A))))

May not terminate!

Prove

Twin(Diddy)
Inference Technique IV: Compilation to Prop. Logic

• Sentence S:
  \( \forall_{\text{city}} a, b \) Connected(a, b)
• Universe
  Cities: seattle, tacoma, enumclaw
• Equivalent propositional formula?
  \( Cst \land Cse \land Cts \land Cte \land Ces \land Cet \)
  \( \forall \) converted to a bunch of \( \land \)'s

Compilation to Prop. Logic (cont)

• Sentence S:
  \( \exists_{\text{city}} c \) Biggest(c)
• Universe
  Cities: seattle, tacoma, enumclaw
• Equivalent propositional formula?
  \( Bs \lor Bt \lor Be \)
  \( \exists \) converted to a bunch of \( \lor \)'s
Compilation to Prop. Logic (cont again)

- **Universe**
  - Cities: seattle, tacoma, enumclaw
  - Firms: IBM, Microsoft, Boeing

- **First-Order formula**
  \[ \forall_{\text{firm } f} \exists_{\text{city } c} \text{ HeadQuarters}(f, c) \]

- **Equivalent propositional formula**
  \[
  \left( (\text{HQis} \lor \text{HQit} \lor \text{HQie}) \land \\
  (\text{HQms} \lor \text{HQmt} \lor \text{HQme}) \land \\
  (\text{HQbs} \lor \text{HQbt} \lor \text{HQbe}) \right)
  \]

Hey!

- You said FO Inference is semi-decidable
- But you compiled it to SAT
  Which is NP Complete
- So now we can always do the inference?!?
  (might take exponential time but still decidable?)

- Something seems wrong here....????
Compilation to Prop. Logic: The Problem

• Universe
  • People: homer, bart, marge
• First-Order formula
  \[ \forall_{\text{people } p} \text{ Male(} \text{FatherOf}(p) \text{)} \]
• Equivalent propositional formula?

\[
\left[ \begin{align*}
(M_{\text{father-homer}} &\land M_{\text{father-bart}} &\land M_{\text{father-marge}} \land \\
(M_{\text{father-father-homer}} &\land M_{\text{father-father-bart}} \land \cdots \\
(M_{\text{father-father-father-homer}} &\land \cdots \\
\cdots
\end{align*}\right]
\]
Not a finite formula

Restricted Forms of FO Logic

• Known, Finite Universes
  Compile to SAT
• Function-Free Definite Clauses (exactly one positive literal, no functions)
  Aka Datalog knowledge bases
• Definite clauses + Inference Process
  E.g., Logic programming using Prolog (uses depth-first backward chaining but may not terminate in some cases)
Hurray! We’ve reached the Midterm mark

Midterm Exam Logistics

• When: Monday, class time
• Where: Here
• What to read: Lecture slides, your notes, Chapters 1-3, 4.1, 5, and 7-9, and practice problems
• Format: Closed book, closed notes except for one 8½” x 11” sheet of notes (double-sided ok)
Friday Class: TA Help Session

- No lecture
- TA Jenn Hanson will be in class 9:30-10:20am to go over some practice problems and answer questions on project or midterm

Midterm Review: Chapters 1 & 2
Agents and Environments

- Browse Chapter 1
- Chapter 2: Definition of an Agent
  Sensors, actuators, environment of agent, performance measure, rational agents

- Task Environment for an Agent = PEAS description
  E.g., automated taxi driver, medical expert
  Know how to write PEAS description for a given task environment
Review: Chapter 2
Agents and Environments

• Properties of Environments
  Full vs. partial observability, deterministic vs. stochastic, episodic vs. sequential, static vs. dynamic, discrete vs. continuous, single vs. multiagent

• Agent Function vs. Agent Program
  State space graph for an agent

• Types of agent programs:
  Simple reflex agents, reflex agent with internal state, goal-based agents, utility-based agents, learning agents

Review: Chapter 3
Search

• State-Space Search Problem
  Start state, goal state, successor function

• Tree representation of search space
  Node, parent, children, depth, path cost g(n)

• General tree search algorithm

• Evaluation criteria for search algorithms
  Completeness, time and space complexity, optimality
  Measured in terms of b, d, and m
Review: Chapter 3
Uninformed Search Strategies

- Know how the following work:
  - Breadth first search
  - Uniform cost search
  - Depth first search
  - Depth limited search
  - Iterative deepening search
- Implementation using FIFO/LIFO
- Completeness (or not), time/space complexity, optimality (or not) of each
- Bidirectional search
- Repeated states and Graph Search algorithm

Review: Chapter 3
Informed Search

- Best-First Search algorithm
  Evaluation function \( f(n) \)
  Implementation with priority queue
- Greedy best-first search
  \( f(n) = \text{heuristic function } h(n) = \text{estimate of cost from } n \text{ to goal} \)
  E.g., \( h_{\text{SLD}}(n) = \text{straight-line distance to goal from } n \)
  Completeness, time/space complexity, optimality
Review: Chapter 3
A* Search

- A* search = best-first search with \( f(n) = g(n) + h(n) \)
- Know the definition of *admissible* heuristic function \( h(n) \)
- Relationship between admissibility and optimality of A*
- Consistent heuristic function
- Completeness, time/space complexity, optimality of A*
- Comparing heuristics: Dominance
- Iterative-deepening A*

Review: Chapter 3 and 4.1
Heuristics & Local Search

- Relaxed versions of problems and deriving heuristics from them
- Combining multiple heuristic functions
- Pattern Databases
- Disjoint pattern databases
- Local search:
  - Hill climbing, global vs. local maxima
    - Stochastic hill climbing
    - Random Restart hill climbing
  - Simulated Annealing
  - Local Beam Search
  - Genetic Algorithms
Review: Chapter 5
Adversarial Search

• Games as search problems
• MAX player, MIN player
• Game Tree, n-Ply tree
• Minimax search for finding best move
  Computing minimax values for nodes in a game tree
  Completeness, time/space complexity, optimality
• Minimax for multiplayer games

Review: Chapter 5
Adversarial Search

• Alpha Beta Pruning
  Know how to prune trees using alpha-beta
  Time complexity
• Fixed Depth (cutoff) search
  Evaluation functions
• Iterative deepening game tree search
• Transposition tables (what? why?)
• Game trees with chance nodes
  Expectimimimax algorithm
Review: Chapter 7
Logical Agents

• What is a Knowledge Base (KB)?
  ASK, TELL
• Wumpus world as an example domain
• Syntax vs. Semantics for a language
• Definition of Entailment
  \( KB \models \alpha \) if and only if \( \alpha \) is true in all worlds where \( KB \) is true.
• Models and relationship to entailment
• Soundness vs. Completeness of inference algorithms

Review: Chapter 7
Logical Agents

• Propositional Logic
  Syntax and Semantics, Truth tables
  Evaluating whether a statement is true/false
• Inference by Truth Table Enumeration
• Logical equivalence of sentences
  Commutativity, associativity, etc.
• Definition of validity and relation to entailment
• Definition of satisfiability, unsatisfiability and relation to entailment
Review: Chapter 7
Logical Agents

• Inference Techniques
  Model checking vs. using inference rules
• Resolution
  Know the definition of literals, clauses, CNF
  Converting a sentence to CNF
  General Resolution inference rule
• Using Resolution for proving statements
  To show $KB \models \alpha$, show $KB \land \neg \alpha$ is unsatisfiable
  by deriving the empty clause via resolution

Review: Chapter 7
Logical Agents

• Forward and Backward chaining
  Know definition of Horn clauses
  AND-OR graph representation
  Modus ponens inference rule
  Know how forward & backward chaining work
• DPLL algorithm
  How is it different from TT enumeration?
• WalkSAT: Know how it works
  Evaluation function, 3-CNF
  $m/n$ ratio and relation to hardness of SAT
Review: Chapter 8
First-Order Logic (FOL)

• First-Order Logic syntax and semantics
  Constants, variables, functions, terms, relations (or predicates), atomic sentences
  Logical connectives: and, or, not, \( \Rightarrow, \Leftrightarrow \)
  Quantifiers: \( \forall \) and \( \exists \)

• Know how to express facts in FOL
  Interaction between quantifiers and connectives
  Nesting of quantifiers

• Interpretations, validity, satisfiability, and entailment

Review: Chapter 9
Inference in FOL

• FOL Inference Techniques
  Universal instantiation
  Existential instantiation
    Skolemization: Skolem constants, Skolem functions
  Unification
    Know how to compute most general unifier (MGU)
  Generalized Modus Ponens (GMP) and Lifting
  Forward chaining using GMP
  Backward chaining using GMP
  Resolution in FOL
    Standardizing apart variables, converting to CNF
  Compilation to Propositional Logic and using SAT solvers
Good luck on the midterm!

Midterm?! I got binders full of wumpuses!