CSE 473

Chapter 9 Wrap-Up
and
Midterm Review

Inference IV:
Compilation to Prop. Logic

• Sentence $S$:
  \[ \forall_{\text{city}} a,b \text{ Connected}(a,b) \]

• Universe
  Cities: seattle, tacoma, enumclaw

• Equivalent propositional formula?

\[ Cst \land Cse \land Cts \land Cte \land Ces \land Cet \]
Compilation to Prop. Logic (cont)

- Sentence $S$:
  $\exists_{\text{city } c} \; \text{Biggest}(c)$
- Universe
  Cities: seattle, tacoma, enumclaw
- Equivalent propositional formula?

$$Bs \lor Bt \lor Be$$

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

$$\left[ (HQ_{\text{is}} \lor HQ_{\text{it}} \lor HQ_{\text{ie}}) \land 
(HQ_{\text{ms}} \lor HQ_{\text{mt}} \lor HQ_{\text{me}}) \land
(HQ_{\text{bs}} \lor HQ_{\text{bt}} \lor HQ_{\text{be}}) \right]$$
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
  \[ \forall_{\text{people } p} \text{ Male(FatherOf}(p)) \]
• Equivalent propositional formula

\[
\begin{align*}
&M_\text{father-homer} \land M_\text{father-bart} \land M_\text{father-marge} ^ \land \\
&M_\text{father-father-homer} \land M_\text{father-father-bart} ^ \land \\
&M_\text{father-father-father-homer} ^ \land ...
\end{align*}
\]
**Restricted Forms of FO Logic**

- **Known, Finite Universes**
  Compile to SAT
- **Function-Free Definite Clauses (exactly one positive literal)**
  Aka Datalog knowledge bases
- **Definite clauses + Inference Process**
  E.g., Logic programming using Prolog (uses depth-first backward chaining) - may not terminate in some cases

**Back To the Wumpus World**

- **Recall description:**
  Squares as lists: [1,1] [3,4] etc.
  Square adjacency as binary predicate.
  Pits, breezes, stenches as unary predicates:
  \[ \text{Pit}(x) \]
  Wumpus, gold, homes as functions:
  \[ \text{Home}(\text{Wumpus}) \]
Back To the Wumpus World

• “Squares next to pits are breezy”:
  \[ \forall x, y, a, b: \]
  \[ \text{Pit}(\{x, y\}) \land \text{Adjacent}(\{x, y\}, \{a, b\}) \Rightarrow \text{Breezy}(\{a, b\}) \]

• “Breezes happen only and always next to pits”:
  \[ \forall a, b \: \text{Breezy}(\{a, b\}) \]
  \[ \exists x, y \: \text{Pit}(\{x, y\}) \land \text{Adjacent}(\{x, y\}, \{a, b\}) \]

What About Our Agent?

• How do we go from knowledge of the world to action in the world?
  \[ \Rightarrow \text{Planning algorithms} \]
  (after Midterm)
Midterm Logistics

• When: Wednesday, class time
• Where: Here
• What to read: Lecture slides, your notes, homework problems, and Chapters 1-4 and 6-9 (skip 5)
• Format: Closed book, closed notes except for one 8 ½" x 11" sheet of notes (double-sided ok)
• Blank sheets will be provided

Watch this space for the Midterm Review Slides

Good luck on the midterm!

Happy Halloween!
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, 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 4
Informed Search

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

Review: Chapter 4
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*
• Completeness, time/space complexity, optimality of A*
• Comparing heuristics: Dominance
• Iterative-deepening A*
Review: Chapter 4
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 6
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 6  
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  
  Quiescent nodes  
- Transposition tables (what? why?)  
- Game trees with chance nodes  
  Expectiminimax 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  
  \[ KB \models \alpha \text{ if and only if } \alpha \text{ is true in all worlds where } KB \text{ is true.} \]  
- 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 $KB \models a$, show $KB \land \neg a$ 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, ⇒, ⇔
  - Quantifiers: ∀ and ∃
- **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