CSE 473

Lecture 14

FOL Wrap-Up and Midterm Review

Resolution in First-Order Logic

- FOL resolution rule:

$$\frac{l_1 \vee \cdots \vee l_k, \qquad m_1 \vee \cdots \vee m_n}{(l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)\theta}$$

  where `Unify`$(l_i, \neg m_j) = \theta$.

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

First-Order Resolution Example

- Given

  $\forall x\ man(x) \Rightarrow human(x)$

  $\forall x\ woman(x) \Rightarrow human(x)$

  $\forall x\ singer(x) \Rightarrow man(x) \vee woman(x)$

  singer(Diddy)

- Prove

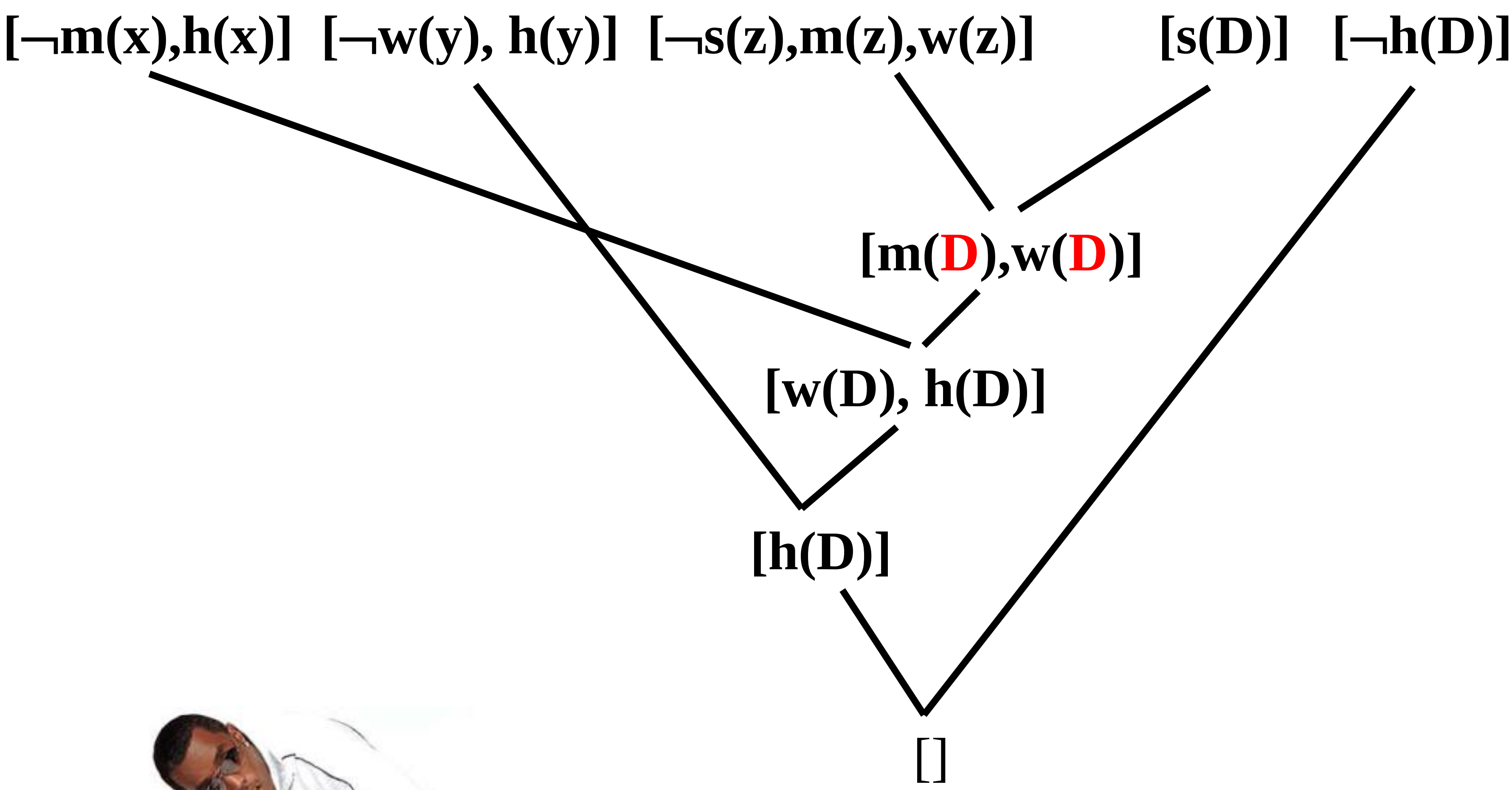
  human(Diddy)

CNF representation (list of clauses standardized apart):

[¬m(x),h(x)] [¬w(y), h(y)] [¬s(z),m(z),w(z)] [s(D)] [¬h(D)]

(The , is shorthand for the OR sign ∨)

FOL Resolution Example



Eh yo homies, dis proves human(Diddy)

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

$\forall x\ \neg[\forall y\ \neg Animal(y) \vee Loves(x,y)] \vee [\exists y\ Loves(y,x)]$

2. Move $\neg$ inwards: $\neg\forall x\ p \equiv \exists x\ \neg p$, $\neg\exists x\ p \equiv \forall x\ \neg p$

$\forall x\ [\exists y\ \neg(\neg Animal(y) \vee Loves(x,y))] \vee [\exists y\ Loves(y,x)]$

$\forall x\ [\exists y\ Animal(y) \wedge \neg Loves(x,y)] \vee [\exists y\ Loves(y,x)]$

Conversion to CNF contd.

3. Standardize variables: Each quantifier uses a different variable

   $\forall x\ [\exists y\ Animal(y) \wedge \neg Loves(x,y)] \vee [\exists z\ Loves(z,x)]$

4. Skolemize: Each existential variable is replaced by a Skolem function of the enclosing universally quantified variables:

   $\forall x\ [Animal(F(x)) \wedge \neg Loves(x,F(x))] \vee Loves(G(x),x)$

5. Drop universal quantifiers:

   $[Animal(F(x)) \wedge \neg Loves(x,F(x))] \vee Loves(G(x),x)$

6. Distribute $\vee$ over $\wedge$ to get CNF (clauses connected by $\wedge$):

   $[Animal(F(x)) \vee Loves(G(x),x)] \wedge [\neg Loves(x,F(x)) \vee Loves(G(x),x)]$

Shorthand:

$[Animal(F(x)),Loves(G(x),x)]\ \ [\neg Loves(x,F(x)),Loves(G(x),x)]$

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:

$\forall x\ American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
$\exists x\ Owns(Nono,x) \wedge Missile(x)$
$\forall x\ Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
$\forall x\ Missile(x) \Rightarrow Weapon(x)$
$\forall x\ Enemy(x,America) \Rightarrow Hostile(x)$
$American(West)$
$Enemy(Nono,America)$

KB in CNF and Resolution

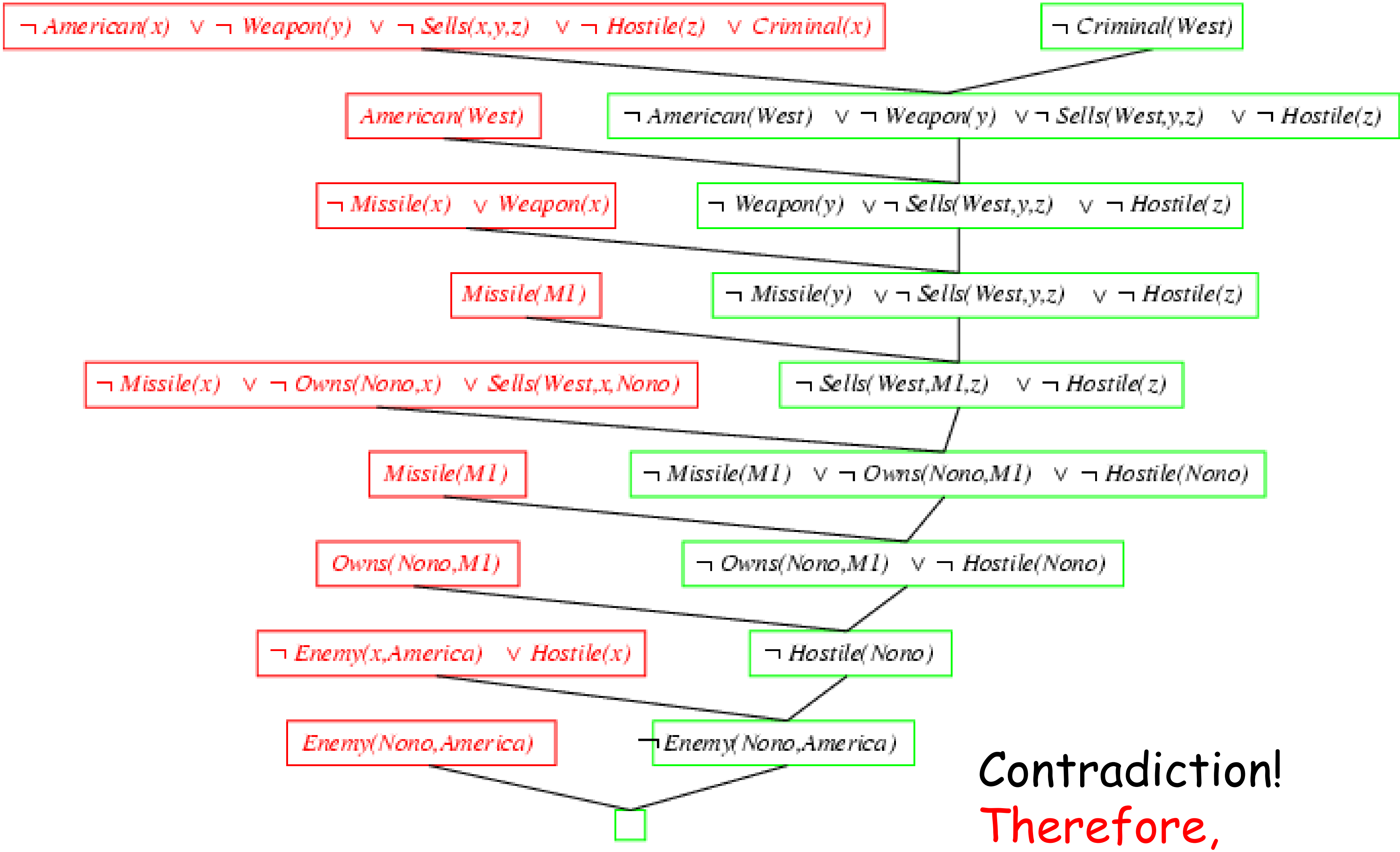
- KB in CNF (note: variables not standardized here)

$\neg$*American(x)* $\vee$ $\neg$ *Weapon(y)* $\vee$ $\neg$ *Sells(x,y,z)* $\vee$ $\neg$ *Hostile(z)* $\vee$ *Criminal(x)*
Owns(Nono, M_1*)* [*Skolem constant* M_1]
Missile(M_1*)*
$\neg$ *Missile(x)* $\vee$ $\neg$ *Owns(Nono,x)* $\vee$ *Sells(West,x,Nono)*
$\neg$ *Missile(x)* $\vee$ *Weapon(x)*
$\neg$ *Enemy(x,America)* $\vee$ *Hostile(x)*
American(West)
Enemy(Nono,America)

- Resolution: Uses "proof by contradiction"
  Show KB $\models$ α by showing KB $\wedge$ $\neg$α unsatisfiable

- To prove Col. West is a criminal, add $\neg$Criminal(West) to KB and derive empty clause

FOL Resolution Example



Contradiction!

Therefore, Criminal(West)

Pitfalls of FOL Resolution

Given

$\forall x \, \exists y \; Twin(x) \Rightarrow Twin(y)$

$Twin(Ashley)$

Prove

$Twin(Diddy)$

$\forall x \; Twin(x) \Rightarrow Twin(F(x))$ **Skolemization**

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

$T(F(A))$

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

$T(F(F(F(A))))$ **May not terminate!**

...

Inference Technique IV: Compilation to Prop. Logic

- Sentence S:

  $\forall_{city}\ a,b$ Connected(a,b)

- Universe

  Cities: seattle, tacoma, enumclaw

- Equivalent propositional formula?

Cst ∧ Cse ∧ Cts ∧ Cte ∧ Ces ∧ Cet

∀ converted to a bunch of ∧'s

Compilation to Prop. Logic (cont)

- Sentence S:

  $\exists_{city}\ c\ \text{Biggest}(c)$

- Universe

  Cities: seattle, tacoma, enumclaw

- Equivalent propositional formula?

  Bs $\vee$ Bt $\vee$ Be

  $\exists$ **converted to a bunch of $\vee$'s**

Compilation to Prop. Logic (cont again)

- Universe
  - Cities: seattle, redmond, everett
  - Firms: Amazon, Microsoft, Boeing
- First-Order formula

$$\forall_{\text{firm}}\, f \quad \exists_{\text{city}}\, c \quad \text{HeadQuarters}(f, c)$$

- Equivalent propositional formula

$$[\ (HQas \lor HQar \lor HQae) \land (HQms \lor HQmr \lor HQme) \land (HQbs \lor HQbr \lor HQbe)\ ]$$

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

  $\forall_{people}\, p \quad \text{Male(FatherOf(}p\text{))}$
- Equivalent propositional formula?

$$[\; (M_{father-homer} \wedge M_{father-bart} \wedge M_{father-marge} \wedge$$
$$(M_{father-father-homer} \wedge M_{father-father-bart} \wedge \ldots$$
$$(M_{father-father-father-homer} \wedge \ldots$$
$$\ldots]$$

Not a finite formula

Restricted Forms of FO Logic

- Known, Finite Universes

  Compile to SAT

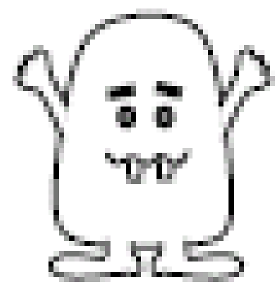
- Function-Free Definite Clauses (exactly one positive literal, no functions)

  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)

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) = 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 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**
- 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
- 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
  - 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 $\models \alpha$ *if and only if* α 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 $KB \wedge \neg\alpha$ is unsatisfiable by deriving the empty clause via resolution

Review: Chapter 7
Logical Agents

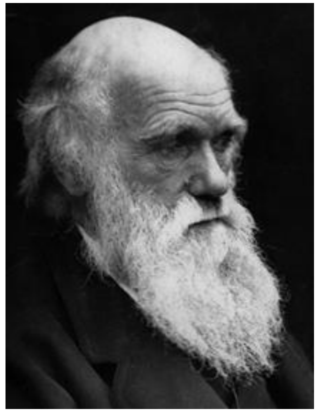
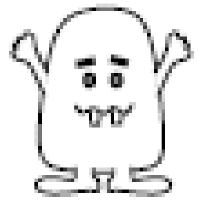
- Forward and Backward chaining
  - Know definition of Horn clauses
  - AND-OR graph representation
  - Modus ponens inference rule
  - Know how forward & backward chaining work
- WalkSAT: Know how it works
  - Evaluation function, 3-CNF
  - m/n ratio and relation to hardness of SAT

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


STAR TREK





Yo! Good luck
on yo midterm!