Logic

Week 4
cs/philo 372

- Logic & AI
- Propositional Logic
- Prolog

Logic & AI

- Logic is the way in which we put together facts.
- One of the 6 forms of human intelligence
- It is the one with the least complex “learning” component.
  - Add a new fact
    - GeorgeBush is President
  - Add a new rule
    - If parent & male then father.
  - Why: because
- What are facts / rules?
  - Watanabe (1969) Theorem of the Ugly Duckling

Logic, AI, and Cyc

- “Cyc is an artificial intelligence project that attempts to assemble a comprehensive ontology and database of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning.”
  - Ontology = rules of relationships between facts
- Started in 1984
- ... vision is to create the world's first true artificial intelligence, having both common sense and the ability to reason with it.

Cyc -- more

- Cyc is – at its core – a deductive theorem prover.
- It has facts and rules that relate those facts
- So, given a question it tries to “prove” the question given the rules and facts it knows.
- Q: Does Lassie have a nose?
  - Lassie is a dog.
  - Dogs are mammals
  - .....

Logic

- Must have
  - Syntax
  - Semantics
  - Reasoning
    - Entailment
      - If A is true, then B must also be true
- Inference Algorithms
  - Soundness
    - Only entailed sentences can be derived
  - Completeness
    - Not always possible because some spaces are infinite
      - e.g., Algebra (ref Hilbert spaces & Godel's incompleteness Theorem)

Propositional Logic

- Contains only
  - Atomic Facts
    - A, B, C
  - Rules linking facts
    - If A & B & C then D
- Simple but rich
  - Consider
    - Geoff is a man
    - Men are HomoSapiens
    - HomoSapiens are mammals
  - Conclude : Geoff is a mammal

The Game of Life
In Propositional Logic

Consider a plane G marked off into a rectilinear grid
The each point in the grid is a propositional fact.
Call these points Ga,b
For each point write the following rules (e.g. for G3,4)

- G3,4 & G2,4 & G4,4 & not G3,3 & not G3,5 => GG3,4
- G3,4 & G2,4 & not G4,4 & G3,3 & not G3,5 => GG3,4
- G3,4 & not G2,4 & G4,4 & G3,3 & not G3,5 => GG3,4
- G3,4 & G2,4 & not G4,4 & not G3,3 & G3,5 => GG3.4
- G3,4 & not G2,4 & G4,4 & not G3,3 & G3,5 => GG3,4
- G3,4 => not GG3,4
- not G3,4 & G2,4 & G4,4 & G3,3 & not G3,5 => GG3,4
- not G3,4 & G2,4 & G4,4 & not G3,3 & G3,5 => GG3,4
- not G3,4 & G2,4 & not G4,4 & G3,3 & G3,5 => GG3,4
- not G3,4 & not G2,4 & G4,4 & G3,3 & G3,3 => GG3,4
- not G3,4 => GG3.4.

Write this set of rules for every point.
To start the game, set a selection of the points Ga,b to true.

The play continues by determining values got all GG. The copy values for all GG back to G. Repeat.

The Game of Life (contd)

- Semantics
  - For each point, go through set of rules until find one that matches, then stop
  - Evaluate the rules for every point in parallel
  - , indicates conjunction
  - Disjunction only through multiple rules
  - Inference
    - Uses modus ponens – given “A => B” then if A true, can conclude B

Prop Logic can be hard

- 3SAT is NP complete
  - Find a set of truth values such that the following is true
  - (A|B|C) & (D|-E|F) & (-G|H|I)...
    - A,B,C are variables
    - | == or
    - & == and
    - All variables can appear multiple times
    - 3SAT is one of the canonical NP-complete problems

Propositional Logic analysis

- A purely “declarative” statement
  - Procedures are relegated to semantics of the logic
  - But many things are more easily phrased propositionally
    - For example, the game of life
- Lack any way to express “unknown”
  - Only true or false. Stuff you do not care about.
- Is “compositional”
  - The whole is exactly the sum of the parts
- Rather wordy

More Propositional Analysis

- Representation of Relations
  - Can sort of be done through rules
    - WutheringHeights <=> authorEmilyBronte & isabook & publishedin1847 & ....
  - Really most of these are relations
    - author(EmilyBronte, WutheringHeights).
    - published(WutheringHeights, 1847).
    - etc

First Order Logic

- Directly address the problem of representing relations
- FOL consists of
  - User defined:
    - Predicates – essentially propositional facts
    - Relations (predicates)
  - Language defined:
    - Forall, thereExists
    - And, or, not, then, if and only if
- So what is second order?

FOL facts

- Terms
  - John
  - Robin
- Unary relations
  - king(John)
  - evil(John)
  - good(Robin)
- Binary Relations
  - fight(John, Robin).
- N-ary relations

Game of Life in FOL

Consider a plane G marked off into a rectilinear grid
The each point in the grid is a propositional fact.
Call these points G(a,b)
For each point write the following rules

- G(a,b) & G(a-1,b) & G(a+1,b) & not G(a,b-1) & not G(a,b+1) => GG(a,b)
- G(a,b) & G(a-1,b) & not G(a+1,b) & G(a,b-1) & not G(a,b+1) => GG(a,b)
- G(a,b) & not G(a-1,b) & G(a+1,b) & G(a,b-1) & not G(a,b+1) => GG(a,b)
- G(a,b) & G(a-1,b) & not G(a+1,b) & not G(a,b-1) & G(a,b+1) => GG(a,b)
- G(a,b) & not G(a-1,b) & G(a+1,b) & not G(a,b-1) & G(a,b+1) => GG(a,b)
- G(a,b) => not GG(a,b)
- not G(a,b) & G(a-1,b) & G(a+1,b) & G(a,b-1) & not G(a,b+1) => GG(a,b)
- not G(a,b) & G(a-1,b) & G(a+1,b) & not G(a,b-1) & G(a,b+1) => GG(a,b)
- not G(a,b) & G(a-1,b) & not G(a+1,b) & G(a,b-1) & G(a,b+1) => GG(a,b)
- not G(a,b) & not G(a-1,b) & G(a+1,b) & G(a,b-1) & G(a,b+1) => GG(a,b)
- not G(A,b) => GG(a,b)

To start the game, set a selection of the points Ga,b to true.

FOL inference

- Given the facts on previous slide
  - king(x)
    - x=John
  - fight(x, Robin)
    - x=john
-

Prolog

- Subset of FOL
  - Allows only “Horn Clauses”
    - A disjunction of literals with at most 1 positive literal
      - A or (not B) or (not C) or (not D)
    - A horn clause can also be written
      - B & C & D -> A
  - Reason for limitation
    - This gives nice readable rules
    - Determining satisfiability over Horn clauses is P-complete.
  - Is this too limiting?

Prolog Basics

- All prolog sentences must end with .
- geoff.
  - This sentence “asserts” the fact “geoff” into prolog
- Geoff.
  - Names with initial cap are variables, so this is illegal, it is an unbounded variable
- prof(geoff).
  - Asserts fact that geoff is a prof

SWI Prolog

- Starting
  - CS department machines: pl
  - On mac: /usr/local/bin/swipl
  - On PC: GUI installed in Program Files/pl
- Stopping
  - mac/unix <ctrl>d
  - Windows: GUI quit
- Loading Files
  - mac/unix   create files name.pl then ['name'].
  - PC – use the menu
- SWI has free installs on unix/mac/PC

Prolog Reasoning

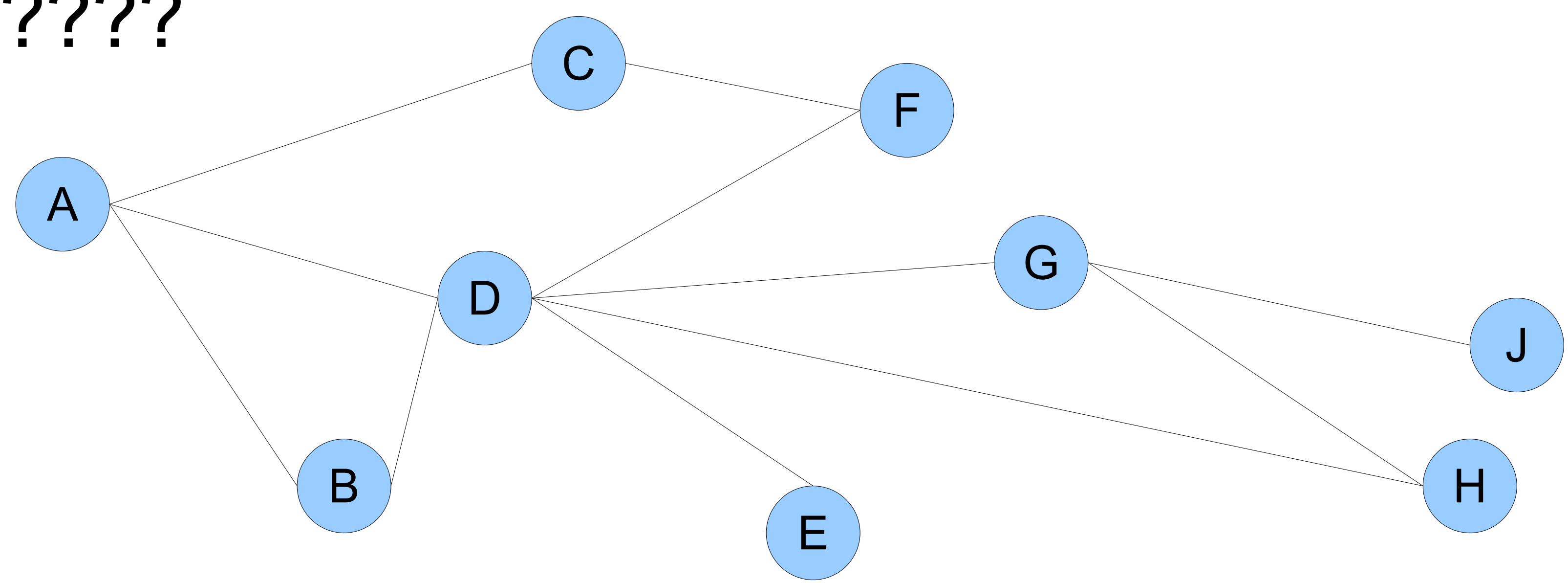
- Suppose you enter the facts:
  - prof(geoff).
  - prof(deepak).
  - prof(diana).
    - In swiprolog facts must be read from a file.
- Then entered the query:
  - prof(X).
    - This is equivalent to asking “what can be bound to X to make this statement true.
    - To see all possible “bindings” of X hit “;” after seeing each
- Problem – this DB is inadequate
  - improve it.

Prolog Rules

- human(X) :- prof(X).
- Prolog reasons with rules in reverse.
  - So this rule says “conclude X is a human if you can show that X is a prof.
- Problem: extend this to show profs are mammals because they are human.

Prolog Reasoning

- Write facts to put in the graph below
- Write rules to determine if there is a path of length 2 connecting 2 points
  - edges are uni-directional left to right
- Write rules that return the intervening node.
- Extend to bi-directional edges without adding facts????



Prolog Lists & output

- Lists are written [a,b,c,d]
- Scanning a list
  - scan([]).
  - scan([Head | Tail]) :- inform(Head), scan(Tail).
  - One of the following rules:
    - inform(X) :- nl, write([hello,X]).
    - inform(X) :- write([hello, X]), nl.
    - inform(X) :- put(X), nl.
    - inform(X) :- nl, put(X).
- Note that “write” uses a prolog list.

Prolog – things to remember

- If you want to know what prolog – or return a value, you need a variable. E.g.,
  - p2(X, A, Y) :- edge(X,A), edge(A,Y).
  - p2(a,Xxxx,d).
    - This would show you the value(s) of Xxxx so that it is a node that links a and d.
- Prolog evaluates in order from the top of file down. So changing order of rules and facts can change program behavior.