Knowledge Representation II
(Inference in Propositional Logic)

CSE 473

Continued...
Last Time

• Inference Algorithms
  As search
  Systematic & stochastic  [[skipped this]]

• Themes
  Expressivity vs.
  Tractability
Special Syntactic Forms: CNF

- **General Form:**
  \[((q \land \neg r) \supset s)) \land \neg (s \land t)\]

- **Conjunction Normal Form (CNF)**
  \[\neg q \lor r \lor s \land (\neg s \lor \neg t)\]

Set notation: \{ \(\neg q, r, s\), \(\neg s, \neg t\) \}
empty clause () = false
Resolution

If the unicorn is mythical, then it is immortal, but if it is not mythical, it is a mammal. If the unicorn is either immortal or a mammal, then it is horned.

Prove: the unicorn is horned.

\[
\begin{align*}
\neg A \lor H & \quad (\neg H) \quad (\neg I \lor H) \\
(M \lor A) & \quad (\neg A) \quad (\neg I) \quad (\neg M \lor I) \\
(M) & \quad (\neg M) \\
() &
\end{align*}
\]

\(M = \) mythical

\(I = \) immortal

\(A = \) mammal

\(H = \) horned
Inference 4: DPLL
(Enumeration of Partial Models)
[Davis, Putnam, Loveland & Logemann 1962]
Version 1

dpll_1(pa) {
  if (pa makes F false) return false;
  if (pa makes F true) return true;
  choose P in F;
  if (dpll_1(pa U \{P=0\})) return true;
  return dpll_1(pa U \{P=1\});
}

Returns true if F is satisfiable, false otherwise
Improving DPLL

If literal $L_1$ is true, then clause $(L_1 \lor L_2 \lor \ldots)$ is true.
If clause $C_1$ is true, then $C_1 \land C_2 \land C_3 \land \ldots$ has the same value as $C_2 \land C_3 \land \ldots$.
Therefore: Okay to delete clauses containing true literals!

If literal $L_1$ is false, then clause $(L_1 \lor L_2 \lor L_3 \lor \ldots)$ has the same value as $(L_2 \lor L_3 \lor \ldots)$.
Therefore: Okay to delete clauses containing false literals!

If literal $L_1$ is false, then clause $(L_1)$ is false.
Therefore: the empty clause means false!
Further Improvements

• Unit Literals

A literal that appears in a singleton clause

{{¬b c}{¬c}{a ¬b e}{d b}{e a ¬c}}

*Might as well set it true! And simplify*

{{¬b} {a ¬b e}{d b}}

{{d}}

• Pure Literals

A symbol that always appears with same sign

{{a ¬b c}{¬c d ¬e}{¬a ¬b e}{d b}{e a ¬c}}

*Might as well set it true! And simplify*

{{a ¬b c} {¬a ¬b e} {e a ¬c}}
DPLL (for real!)
Davis – Putnam – Loveland – Logemann

dplll(F, literal) {
    remove clauses containing literal
    shorten clauses containing \( \neg \)literal
    if (F contains no clauses) return true;
    if (F contains empty clause)
        return false;
    if (F contains a unit or pure L)
        return dplll(F, L);
    choose V in F;
    if (dplll(F, \( \neg \)V)) return true;
    return dplll_2(F, V);
}
DPLL (for real)

\[(\neg a \lor c) \land (a \lor \neg b) \land (a \lor \neg c)\]

Diagram:
- Node a connected to nodes b and c.
- Node b connected to node c.
- Node c has a green leaf node.
Heuristic Search in DPLL

• Heuristics are used in DPLL to select a (non-unit, non-pure) proposition for branching

• Idea: identify a most constrained variable
  Likely to create many unit clauses

• MOM's heuristic:
  Most occurrences in clauses of minimum length
Success of DPLL

- 1962 – DPLL invented
- 1992 – 300 propositions
- 1997 – 600 propositions (satz)
- Additional techniques:
  Learning conflict clauses at backtrack points
  Randomized restarts
  2002 (zChaff) 1,000,000 propositions – encodings of hardware verification problems
Outline

• **Inference Algorithms**
  As search
  Systematic & stochastic

• **Themes**
  Expressivity vs.
  Tractability
WalkSat

- Local search over space of complete truth assignments
  - With probability $P$: flip any variable in any unsatisfied clause
  - With probability $(1-P)$: flip best variable in any unsat clause
    - Like fixed-temperature simulated annealing

- SAT encodings of N-Queens, scheduling
- Best algorithm for random K-SAT
  - Best DPLL: 700 variables
  - Walksat: 100,000 variables
Random 3-SAT

• Random 3-SAT sample uniformly from space of all possible 3-clauses
  \( n \) variables, \( l \) clauses

• Which are the hard instances? around \( l/n = 4.3 \)
Random 3-SAT

- Varying problem size, $n$

- Complexity peak appears to be largely invariant of algorithm
  backtracking algorithms like Davis-Putnam
  local search procedures like GSAT

- What’s so special about 4.3?
Random 3-SAT

- Complexity peak coincides with solubility transition

\[
\frac{l}{n} < 4.3 \text{ problems under-constrained and SAT}
\]

\[
\frac{l}{n} > 4.3 \text{ problems over-constrained and UNSAT}
\]

\[
\frac{l}{n}=4.3, \text{ problems on “knife-edge” between SAT and UNSAT}
\]
Real-World Phase Transition Phenomena

- Many NP-hard problem distributions show phase transitions -
  - job shop scheduling problems
  - TSP instances from TSPLib
  - exam timetables @ Edinburgh
  - Boolean circuit synthesis
  - Latin squares (alias sports scheduling)

- Hot research topic: predicting hardness of a given instance, & using hardness to control search strategy (Horvitz, Kautz, Ruan 2001-3)
Restricted Expressiveness

• 2-SAT

• Horn Theories
Horn Theories

• Horn clause $\equiv$ at most one positive literal
  \[
  \{ (\neg q \lor \neg r \lor s), (\neg s \lor \neg t) \}
  \{ ((q \land r) \supset s), ((s \land t) \supset false) \}
  \]

• Many problems naturally take the form of such if/then rules
  
  If (fever) AND (vomiting) then FLU

• Unit propagation is refutation complete for Horn theories
  
  Good implementation - linear time!
Summary: Algorithms

- Forward Chaining
- Resolution
- Model Enumeration
- Enumeration of Partial Models (DPLL)
- Walksat
Themes

• Expressiveness
  Expressive but awkward
  No notion of objects, properties, or relations
  Number of propositions is fixed

• Tractability
  NPC in general
  Completeness / speed tradeoff
  Horn clauses, binary clauses