COS402- Artificial Intelligence
Fall 2015

Lecture 8: Applications of solving CNF

Outline

- **Brief review on search techniques**
  - **Blind search, heuristic search, and game search**
- **Brief review on logical inference**
  - **Propositional logic, model checking, and theorem proving**
- **Applications of solving CNF**
  - **Many problems can be reduced to SAT problems**

5 components of search problems

- **Initial state**
- **Actions**
- **Transitional model**
- **Goal test**
- **Path cost**

Blind search

- **Breadth-First Search (BFS)**
- **Depth-First Search (DFS)**
- **Depth-Limited Search (DLS)**
  - **The depth of the root node is 0.**
- **Iterative-Deepening Search (IDS)**
  - **Start at l = 0.**
- **Bidirectional Search**

Heuristic Search

- **Admissible and consistent heuristics**
- **Greedy-First Search**
  - **f(n) = h(n)**
- **A* Search**
  - **f(n) = g(n)+h(n)**
  - **A* graph search is optimal when using consistent heuristics**
  - **A* tree search is optimal when using admissible heuristics**

Search in Games

- **Games**
  - **2 player**
  - **Zero-sum**
- **The Minimax algorithm**
  - **Complete and optimal**
- **Alpha-beta pruning**
  - **Significantly reduce the number of nodes searched while maintaining the optimality of the Minimax algorithm.**

Logical inference

- **Problem: Can we infer a new fact given a set of known facts (KB |= α ?)**
- **Propositional logic**
  - **Propositional symbols, Syntax and semantics**
- **Model checking**
  - **DPLL**
  - **WALKSAT**
- **Theorem proving**
  - **Resolution algorithm (is KB ∧ ~α unsatisfiable?)**
  - **Forward/backward chaining (KB: Horn clauses, α : single positive symbol)**

DPLL and WALKSAT

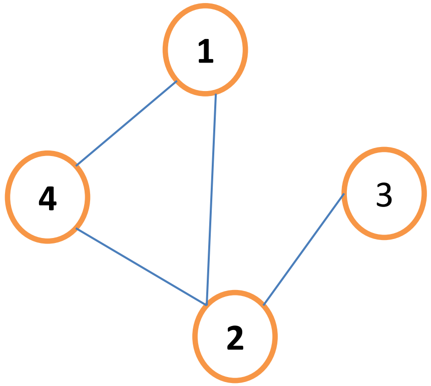
- **DPLL**
  - **Complete and sound**
  - **Determine KB |= α**
  - **Check satisfiability of a cnf + find a model if it is satisfiable**
- **WALKSAT**
  - **Sound, but not complete**
  - **Mostly used for finding a model when a cnf is satisfiable**

Applications of solving CNF

- **SAT is used in problems other than logical inference**
  - **N-queen problem**
  - **3-coloring graph**
  - **Hamiltonian path**
  - **Planning**

Reduce 3-coloring graph to SAT

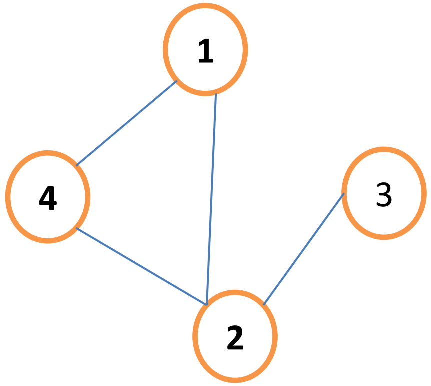
- **Define Symbols:**
  - **P_{ij} : node i is colored in color j**
  - **i = 1,2,3 or 4**
  - **j = r, g or b**



- **Express facts/rules in clauses**
  1. **Each node gets one color**
  2. **Two nodes sharing a common edge can’t be colored the same**

Reduce 3-coloring graph to SAT

1. Each node gets one color



(1) Each node gets at least one color

$P_{1r} \vee P_{1g} \vee P_{1b}$

$P_{2r} \vee P_{2g} \vee P_{2b}$

$P_{3r} \vee P_{3g} \vee P_{3b}$

$P_{4r} \vee P_{4g} \vee P_{4b}$

(2) Each node gets only one color

$(\sim P_{1r} \vee \sim P_{1g}) \wedge (\sim P_{1r} \vee \sim P_{1b}) \wedge (\sim P_{1g} \vee \sim P_{1b})$

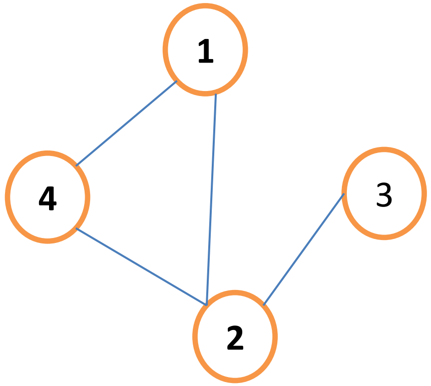
$(\sim P_{2r} \vee \sim P_{2g}) \wedge (\sim P_{2r} \vee \sim P_{2b}) \wedge (\sim P_{2g} \vee \sim P_{2b})$

$(\sim P_{3r} \vee \sim P_{3g}) \wedge (\sim P_{3r} \vee \sim P_{3b}) \wedge (\sim P_{3g} \vee \sim P_{3b})$

$(\sim P_{4r} \vee \sim P_{4g}) \wedge (\sim P_{4r} \vee \sim P_{4b}) \wedge (\sim P_{4g} \vee \sim P_{4b})$

Reduce 3-coloring graph to SAT(cnt'd)

2. Two nodes sharing a common edge can't be colored the same



- **For edge 1-4**

  $(\sim P_{1r} \lor \sim P_{4r}) \land (\sim P_{1g} \lor \sim P_{4g}) \land (\sim P_{1b} \lor \sim P_{4b})$

- **For edge 2-4**
  - **$(\sim P_{2r} \lor \sim P_{4r}) \land (\sim P_{2g} \lor \sim P_{4g}) \land (\sim P_{2b} \lor \sim P_{4b})$**

- **For edge 1-2**
  - **$(\sim P_{1r} \lor \sim P_{2r}) \land (\sim P_{1g} \lor \sim P_{2g}) \land (\sim P_{1b} \lor \sim P_{2b})$**

- **For dege 2-3**
  - **$(\sim P_{2r} \lor \sim P_{3r}) \land (\sim P_{2g} \lor \sim P_{3g}) \land (\sim P_{2b} \lor \sim P_{3b})$**

---Put all clauses in a cnf and pass to a sat-solver.

---A model for the constructed cnf is a solution to the original problem.

---Legal coloring is guaranteed by the rules in 1 and 2.

Announcement & Reminder

- **P1 is due today**

  --- due by midnight, upload your files to CS dropbox.

  --- remember to press the "check all submitted files" button. No credit will be given the code that does not compile.

- **P2 has been released and is due on Tuesday Oct. 27th**

  --- due by midnight, upload your files to CS dropbox.