Midterm Review (CMSC 471/671, Fall 2000)

- **State Space**
  - States: initial, goal.
  - State transition rules/operators/actions and costs associate with operations
  - State space as directed graph (nodes, arcs, parents/children)
  - Node generation and node expansion: open/closed nodes, open/closed lists.
  - Solution, solution path and its cost.
  - Be able to represent simple problem-solving as state space search

- **Uninformed (blind) Search Methods**
  - Breadth-first.
  - Depth-first, Depth-limited (plus back-tracking).
  - IDDF: Iterative-deepening depth-first. (motivation, advantage over BF and DF methods.)
  - Uniform-cost search.
  - Bi-directional search
  - Algorithm, time and space complexities, optimality and completeness of each of these search methods

- **Informed Search methods**
  - Evaluation function f(n),
  - Heuristic estimate function h(n)
    - what does h(n) estimate
    - admissible h(n), null h(n), perfect h*(n), more informed h(n)
    - idea of automatic generation of h functions
  - Algorithm A and A*
    - f(n) = g(n) + h(n): what does each of the terms stand for;
    - algorithm (maintaining open/closed lists, delayed termination test; node expansion and generation, handling duplicate nodes, back pointers);
    - difference between algorithms A and A*
    - time and space complexity, completeness and optimality of A*
    - be able to apply A* to simple problems.
  - Ways to improving A* search
    - IDA* (basic idea; how to set f_limit at each iteration; advantages over A*)
    - Dynamic weighting (an algorithm)
    - Pruning open list by f+ (where f+ is the cost of any known solution)
  - Best-first search
  - Greedy search and hill-climbing (algorithms; time and space complexity, completeness and optimality)
  - Basic ideas of other incremental improving search methods
  - Simulated annealing; Genetic algorithm

- Game-Tree Search
  - Perfect 2-player games
  - Game tree (Max and Min nodes; terminal and leave nodes)

- What to search for (one move for Max)
- Heuristic evaluation function f(n) (merit of a board)
- Minimax rule for game tree search
- Idea of alpha-beta pruning, its time and space complexities.
- Difference between general state space search and game tree search
- Be able to apply Minimax rule and alpha-beta pruning to simple problems.

- Propositional Logic (PL)
  - Syntax
    - Propositions
    - Symbols (T, F, proposition symbols)
    - Connectives
    - Definition of PL sentences
  - Semantics
    - Interpretation (an assignment of truth values to all prepositional symbols); models
    - Truth tables for logical connectives
    - Valid (tautology), satisfiable and inconsistent (contradiction) sentences
    - Logical consequence or theorem (S |= X)
  - Equivalence laws
    - P Q iff they have the same truth tables
    - P => Q ~P v Q; distribution /associative/communicative laws, De Morgan's laws
  - Deductive rules
    - Derivation using inference rules: S |-- X
    - Modus Ponens, Modus Tollens, Chaining, And Introduction, And Elimination, etc.
    - Resolution rule (and CNF)
  - Deductive inference
    - Using truth table (S |= X iff S => X is valid)
    - Proof procedure (using inference rules)
    - Sound inference rules and proof procedures (if S |-- X then S |= X)
    - Complete proof procedures (if S |= X then S |-- X). (exponential time complexity)
  - Treating PL inference as state space search

- First Order Logic (FOL)
  - Syntax
    - Terms (constants, variables, functions of terms)
    - Predicates (special functions, ground predicates), atoms and literals
    - Logical connectives
    - Quantifiers (universal and existential), their scopes, De Morgan's law with quantifiers
    - Definitions of FOL sentences and well-formed formulas (wffs)
  - Semantics
    - Interpretation (constants, functions, and predicates) and models
    - Semantics of logical connectives and quantifiers
    - Valid, satisfiable, and inconsistent sentence(s)
    - Logical consequences
    - Be able to translate between English sentences and FOL sentences