CS472
Foundations of Artificial Intelligence
Midterm Exam Solution

October 26, 1998

1. **State Space Search: (15 points total)**

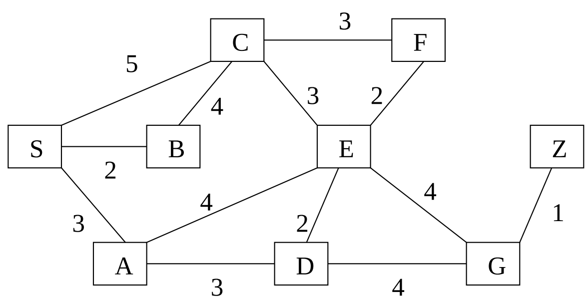

Figure 1: Graph for Heuristic Search question.

See the graph in Figure 1. The start state is S and the goal state is Z. The numbers on the arcs indicate the cost of traversing that arc. Whenever the search algorithm requires a heuristic estimating function, use the following:

n	S	A	B	C	D	E	F	G	Z
h(n)	8	7	6	5	4	3	2	1	0

(a) (5 pts.) Using the graph in Fig. 1 and **uniform-cost search**: (1) list the nodes in the order they would be expanded; (2) list the nodes that lie along the final correct path to the goal.

(1) $S, B, A, C, D, E, F, G, Z$.
(2) S, A, D, G, Z.

(b) (5 pts.) Same as (a), but using **greedy search**.

(1) S, C, F, E, G, Z
(2) S, C, F, E, G, Z

(c) (5 pts.) Same as (a), but using **A* search**.

(1) $S, B, A, D, C, E, F, G, Z$
(2) S, A, D, G, Z

2. α-β **Pruning (20 pts)**

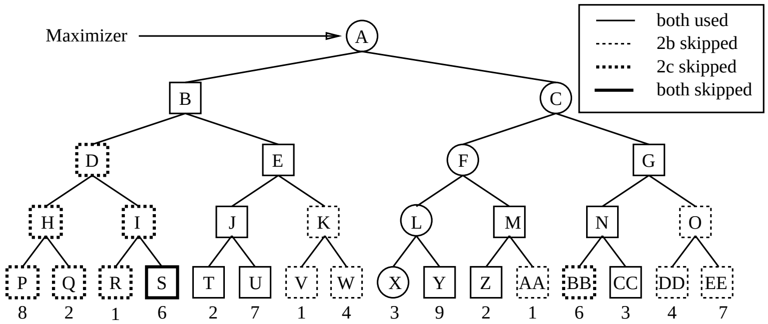

Figure 2: Game Tree for Alpha-beta Pruning

These questions refer to the game tree in Figure 2.

(a) **(5 pts.)** What is the solution? That is, which move should be made next and what is the expected value of that move?

The correct move is to node C, with a value of 3.

(b) **(10 pts.)** Using alpha-beta pruning (and standard left-to-right evaluation of nodes), how many of the leaves get evaluated? Indicate all parts of the tree that are cut off. Indicate the winning path or paths. Strike out all static evaluation values that do not need to be computed.

10 leaves get evaluated; they are $P, Q, R, T, U, X, Y, Z, BB$, and CC.
The winning path is $A \rightarrow C \rightarrow F \rightarrow L \rightarrow X$.

(c) **(5 pts.)** Explain why searches in game-playing programs generally go forward from the current position instead of backward from the goal.

First off, there may be many possible goal states, so backward-search would have to either choose (with the possibility of choosing wrong) or try them all. Many of these goals may be unattainable, so searching from them is a waste of time. Finally, in many cases the distance from any goal node is so far that there would not be enough resources to search back to the current state.

3. **CSP - The Missionary & Cannibals Puzzle (25 pts. total)**

Three missionaries and three cannibals are on the left side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the right side of the river, without leaving a group of missionaries in one place outnumbered by the cannibals in that place.

Consider the following set of variables, defining operators (moves) indexed by time:

(a) $O1MLR_i$ — move one missionary from the left to the right side at time i.

(b) $O2MLR_i$ — move two missionaries from the left to the right side at time i.

(c) $O1CLR_i$ — move one cannibal from the left to the right side at time i.

(d) $O2CLR_i$ — move two cannibals from the left to the right side at time i.

(e) $O1M1CLR_i$ — move one missionary and one cannibal from the left to the right side at time i.

(f) $O1MRL_i$ — move one missionary from the right to the left side at time i.

(g) $O2MRL_i$ — move two missionaries from the right to the left side at time i.

(h) $O1CRL_i$ — move one cannibal from the right to the left side at time i.

(i) $O2CRL_i$ — move two cannibals from the right to the left side at time i.

(j) $O1M1CRL_i$ — move one missionary and one cannibal from the right to the left side at time i.

The domain of these operator variables is $\{0, 1\}$. (1 if the operator applies at time i; 0 if not.)

Consider also the state variables (again indexed by time i):

- CL_i — number of cannibals on the left side at time i
- ML_i — number of missionaries on the left side at time i
- CR_i — number of cannibals on the right side at time i
- MR_i — number of missionaries on the right side at time i

The domain of these state variables is $\{0, 1, 2, 3\}$.

Finally, we have two more state variables indicating the position of the boat:

- BL_i — the boat is on the left side at time i.
- BR_i — the boat is on the right side at time i.

The domain of these state variables is $\{0, 1\}$, e.g., if $BL_i = 1$, then the boat is on the left side at time i.

The main decision variables of our problem are the sequence of moves (operators) that have to be performed in order to successfully move all the cannibals and missionaries from one side to the other. In other words, at each time i, an operator, from the set of 10 operators listed above should be selected (*i.e.*, its value should be 1). State variables are used to guarantee that the constraints of the problem are not violated. Let us consider a formulation in which we search for a solution of some fixed length k, i.e., the solution (if found) consists of a chain of k operator applications. This means the time index i goes from 1 to $k + 1$. The initial state is defined by $i = 1$ and the final state by $i = k + 1$.

This problem can then be formulated as a constraint satisfaction problem. In order to do so, constraints must be defined. In this case constraints can be defined as simple equations. For example:

$$BL_i + BR_i = 1$$

This constraint states that, at any time i, the boat is either on the left side or on the right side. (We don't explicitly model the boat in transition. Note that "+" is standard addition.)

(5 pts.) Say in words what the following constraint means:

$$O1MLR_i + O2MLR_i + O1CLR_i + O2CLR_i + O1M1CLR_i + O1MRL_i + O2MRL_i + O1CRL_i + O2CRL_i + O1M1CRL_i = 1 \quad \text{for } i = 1, \cdots, k.$$

The above constraint simply means that **exactly one** operator (move) can be performed at time i.

Using the notation defined above for the variables, write down the following constraints:

- (4 pts.) State what holds in the initial state ($i = 1$).

  $CL_1 = 3, ML_1 = 3,$
  $CR_1 = 0, MR_1 = 0,$
  $BL_1 = 1, BR_1 = 0$

- (4 pts.) In order to move people from the left to the right side at time i, the boat has to be on the left side.

  $O1MLR_i + O2MLR_i + O1CLR_i + O2CLR_i + O1M1CLR_i = BL_i \quad \text{for } i = 1, \cdots, k$
  [We'll have a similar constraint to guarantee the boat is on the right side when moving people from the right side. You don't have to state this constraint.]

- (4 pts.) The number of missionaries on left side at time $(i + 1)$ is the result of the operator selected at time i and the number of missionaries on left side at time i. (Hint: the constraint can be stated as one equation.)

  $ML_{i+1} = ML_i + O1MRL_i + 2(O2MRL_i) + O1M1CRL_i - O1MLR_i - 2(O2MLR_i) - O1M1CLR_i \quad \text{for } i = 1, \cdots, k - 1$
  [We'll have similar constraints for updating the number of cannibals and for the numbers on the right hand side. You don't have to state these constraints.]

- (4 pts.) Are there any other constraints required for the formulation of this problem? If your answer is yes, formulate them.

  Yes, we need to make sure that a group of missionaries in one place is not outnumbered by the cannibals in that place. That is,
  $ML_i \geq CL_i \quad \text{if } ML_i > 0; \quad MR_i \geq CR_i \quad \text{if } MR_i > 0 \quad \text{for } i = 1, \cdots, k$
  For simplicity, it is equivalent to say,
  $ML_i \cdot ML_i \geq ML_i \cdot CL_i; \quad MR_i \cdot MR_i \geq MR_i \cdot CR_i \quad \text{for } i = 1, \cdots, k$
  We also add:
  $ML_i \geq 0; \quad MR_i \geq 0$
  $CL_i \geq 0; \quad CR_i \geq 0$

- (4 pts.) In logical formulations one needs to add frame axioms. Explain briefly what frame axioms are. Do we need explicit frame axioms here? Explain your answer.

  Axioms that describe how the world stays the same (as opposed to how it changed) are called frame axioms.
  We don't need explicite frame axioms here because the above formulation has provided a complete description of how the world evolves.

4. **First-Order Logic (20 pts. total)**

   Consider the following sentences:

   (a) All the people in room B11 are either professors or students.
   (b) If a student is in room B11, then some advisor of his or her is also in room B11.
   (c) The advisor of any student is a professor.
   (d) Someone is in room B11.

   Let $\mathcal{L}$ be a first-order language containing the following predicates:

   - $P(X)$ — X is a professor.
   - $S(X)$ — X is a student.
   - $A(X, Y)$ — X advises Y.
   - $R(X)$ — X is in room B11.

   (a) (8 pts.) State each of the above sentences in $\mathcal{L}$.
      i. $\forall X\ R(X) \Rightarrow P(X) \vee S(X)$
      ii. $\forall X\ R(X) \wedge S(X) \Rightarrow \exists Y\ R(Y) \wedge A(Y, X)$
      iii. $\forall X, Y\ S(X) \wedge A(Y, X) \Rightarrow P(Y)$
      iv. $\exists X\ R(X)$

   (b) (12 pts.) Using resolution, show that "Some professor is in room B11" can be proven from the other sentences. You must use clausal form, show the Skolemized form when needed, and each resolution step n the path to the null clause.

   The correct solution to the problem involves the following steps:

   (a) Express the goal in first order logic:

   $$\exists X\ R(X) \wedge P(X)$$

   Translate the negation of the goal in CNF (negate and give up universal quantifier):

   $$\neg R(X) \vee \neg P(X) \tag{1}$$

   (b) Translate the first order formulas at point a) in CNF.
   To translate i) we just drop the universal quantifier and transform implication in disjunction $((p \Rightarrow q) \Leftrightarrow (\neg p \vee q))$:

   $$\neg R(X) \vee P(X) \vee S(X) \tag{2}$$

   To translate ii) we first use skolemnization to eliminate the existential quantifier on Y by replacing Y with $\mathcal{Y}(X)$, a function that for every X returns an object. After that we drop universal quantifiers and transform implication. We get two clauses in CNF for this first order formula

$$\neg R(X) \vee \neg S(X) \vee R(\mathcal{Y}(X)) \tag{3}$$

$$\neg R(X) \vee \neg S(X) \vee A(\mathcal{Y}(X), X) \tag{4}$$

To translate iii) we eliminate implication and drop the universal quantifiers:

$$\neg S(X) \vee \neg A(Y, X) \vee P(Y) \tag{5}$$

To translate iv) we skolemnize X by replacing it with an object $\mathcal{X}$:

$$R(\mathcal{X}) \tag{6}$$

(c) From now on we will use only resolution and unification on two of the clauses to produce a new one until we produce Λ (the empty clause).

$$(6,1) \quad \neg P(\mathcal{X}) \tag{7}$$

$$(4,5) \quad \neg R(X) \vee \neg S(X) \vee P(\mathcal{Y}(X)) \tag{8}$$

$$(3,1) \quad \neg R(X) \vee \neg S(X) \vee \neg P(\mathcal{Y}(X)) \tag{9}$$

$$(8,9) \quad \neg R(X) \vee \neg S(X) \tag{10}$$

$$(10,2) \quad \neg R(X) \vee P(X) \tag{11}$$

$$(6,11) \quad P(\mathcal{X}) \tag{12}$$

$$(7,12) \quad \Lambda \tag{13}$$

5. **Inference / CSP (20 pts. total)**

(a) **8 pts.** Show that the satisfiability of a 2SAT (i.e., propositional CNF with 2 literals per clause) formula can be determined in polynomial time. Hint: Consider resolution for determining satisfiability. What property of resolution do you exploit? Why does it run in polynomial time?

For a 2SAT formula with n variables, let's consider $P \vee Q$ and $\neg Q \vee R$, which gives $P \vee R$ by resolution. In other words, the clauses introduced by resolution contain at most 2 literals. Since there are at most $C(n, 2)$ number of combinations for n variables, it takes at most $O(n^2)$ resolutions to determine the satisfiability of a 2SAT. Therefore, the satisfiability of a 2SAT can be determined in polynomial time.

(b) **2 pts.** Why does the argument in a) **not** work for 3SAT?

For a 3SAT formula, let's consider $P1 \vee P2 \vee Q$ and $\neg Q \vee R1 \vee R2$, which gives $P1 \vee P2 \vee R1 \vee R2$ by resolution. The clauses introduced by resolution may contain more than 3 literals (after a few steps, clauses can contain up to n literals), and thus the number of resolutions needed to determine the satisfiability of a 3SAT can grow exponentially in n. In short, the argument in a) doesn't work for 3SAT.

(c) **10pts.**

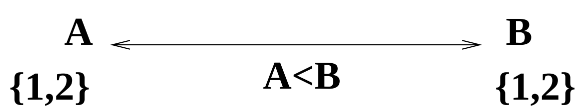

Figure 3: Constraint problem.

Consider the constraint satisfaction problem in figure 3: two variables A and B each with domain $\{1, 2\}$ and a single constraint $A < B$. After applying arc consistency, the domain for A becomes $\{1\}$ and for B $\{2\}$. Explain why.

From the figure it is suggestive that A should be less than B in any case and hence we only have values $\{1\}$ for A and $\{2\}$ for B which would satisfy the arc consistency.

Use resolution to demonstrate that the result of arc consistency is a logically sound for this example. Formalize the problem by introducing propositional variables of the form $A1$ (denoting $A = 1$), $A2$ (denoting $A = 2$), etc. Give the clauses representing that the original domain of A is $\{1, 2\}$ and of B is $\{1, 2\}$.

$$(A1 \vee A2) \tag{14}$$

$$(B1 \vee B2) \tag{15}$$

$A < B$ is captured by several clauses. One example is $(\neg A2 \vee \neg B2)$ (i.e., $A2 \rightarrow \neg B2$). Write down the other clauses.

$$(\neg A1 \vee \neg B1) \tag{16}$$

$$(\neg A2 \vee \neg B1) \tag{17}$$

$$(\neg B1 \vee \neg A1) \tag{18}$$

$$(\neg B1 \vee \neg A2) \tag{19}$$

$$(\neg B2 \vee \neg A2) \tag{20}$$

Finally, show that the clauses capturing the constraint problem logically imply $A1$ and $B2$ by using resolution.

$$(15, 19) \quad (B2 \vee -A2) \tag{21}$$

$$(20, 21) \quad -A2 \tag{22}$$

$$(14, 22) \quad A1 \tag{23}$$

$$(14, 16) \quad (A1 \vee -B1) \tag{24}$$

$$(18, 24) \quad -B1 \tag{25}$$

$$(15, 25) \quad B2 \tag{26}$$