CS 188 Midterm
11:15am–12:30pm March 22, 1990

Please try to be precise in your answers. The maximum possible score on this exam is 100 points. Good Luck!

1 Resolution Proof [20 pts]

Formulate as predicate-calculus expressions the facts given in the following puzzle. Use the resolution method to prove that Cafe-Stanford is criminal. Remember to convert to Conjunctive Normal Form before starting the proof!

1. It is a crime to sell a bad cappucino.

2. All the beverages that Tom drank were purchased by him from either Cafe-Stanford or Cafe-Berkeley.

3. One of the beverages that Tom drank was a bad cappucino.

4. Tom never bought anything from Cafe-Berkeley.

Use the following predicates: Bad (x), Cappucino (x), Criminal (x), Beverage (x), Drank (x, y) for “x drank y”, Sells (x, y, z) for “x sells y to z”.

2 Clause Form [20 pts]

Transform the negation of the following well-formed formula to clause form. Is the ( original ) formula valid?

\[ \exists x(p_1(x) \land q_1(x)) \Rightarrow [\exists x(p_1(x)) \land \exists x(q_1(x))] \]

3 Search [20 pts]

1. A knight on an infinite chessboard must be transferred from an initial position of (0, 0) to a goal position \((m, n)\) using the minimum number of legal knight moves. Find an admissible heuristic function \(h\) that you could use to solve the problem using the \(A^*\) algorithm.
2. Suppose that for some search problem for which you want to use A* search you have found an evaluation function that never overestimates the cost to a goal state by more than K units. How can you get a guaranteed optimal solution from A* search?

4  Lisp [20 pts]

Define a LISP function ALLSUB \( (u, v) \) that returns a list of all occurrences of a list of atoms \( u \) as a sublist of another list of atoms \( v \). The occurrence of a particular sublist is represented by a number \( n \) corresponding to the position in the list \( v \) of the beginning of that occurrence. For example

\[
\text{ALLSUB \('(A A)\, '(A A A B A A)\) = (1 2 5)}
\]

5  Alpha–Beta Search [20 pts]

Explore the tree using the alpha-beta procedure. Assume that the top level is a maximizing level. Cross out all nodes where static evaluation need not occur. Indicate the winning path or paths.