Artificial Intelligence I

A simple game works as follows. We have a board divided into \( n \) by \( m \) square cells. We also have an unlimited number of L-shaped tiles, each made to cover exactly three squares. The tiles can appear in any of the four possible orientations. Our aim is to cover the board completely with non-overlapping tiles.

(a) A single tile on the board can be described using a list such as \([[[1, 1], [1, 2], [2, 1]]]\) containing three tuples, specifying the position of each part of the tile on the board. Consider the following Prolog predicate, which is true if the six variables describe a correct, L-shaped tile.

\[
\text{tile}([\textcolor{red}{[A,B] \textcolor{blue}{, [C,D]} \textcolor{green}{, [E,F]}]}]) \leftarrow \textcolor{red}{C} \text{ is } A+1, \textcolor{blue}{D} \text{ is } B, \text{ E is } A, \text{ F is } B-1; \\
\text{C is } A+1, \text{ D is } B, \text{ E is } A, \text{ F is } B+1; \\
\text{C is } A-1, \text{ D is } B, \text{ E is } A, \text{ F is } B+1; \\
\text{C is } A-1, \text{ D is } B, \text{ E is } A, \text{ F is } B-1. \\
\]

Explain what happens in response to a query of the form

\[
\text{tile}([\textcolor{red}{[4,5]} \textcolor{blue}{, [B,C]} \textcolor{green}{, [D,E]}]).
\]

Keep in mind the effects of backtracking. [2 marks]

(b) Write a Prolog predicate \( \text{goodplace}([\textcolor{red}{[A,B]} \textcolor{blue}{, [C,D]} \textcolor{green}{, [E,F]}],[N,M]) \) that is true if \([\textcolor{red}{[A,B]} \textcolor{blue}{, [C,D]} \textcolor{green}{, [E,F]}]\) is a validly shaped tile and all of its parts lie within an \( N \) by \( M \) board. Your predicate should behave under backtracking in such a way that the response to a query of the form

\[
\text{goodplace}([\textcolor{red}{[10,4]} \textcolor{blue}{, [B,C]} \textcolor{green}{, [D,E]}],[10,10]).
\]

is to find the unspecified values for all tiles which have a valid shape and fall within the board. In this example there would be two such tiles. [6 marks]

(c) Write a Prolog predicate \( \text{tiling}(\text{Available}, \text{Solution}, \text{Size}) \). Here, \( \text{Size} \) is the size of the board represented as above, \( \text{Solution} \) is a list of tiles that solves the problem, and \( \text{Available} \) is a list of available positions on a board of the given size. For example, if \( \text{Size} \) is \([2,2]\) then \( \text{Available} \) is \([[[1,1],[1,2],[2,1],[2,2]]]\).

Your predicate should be true if the \( \text{Solution} \) given is a valid one, and should be capable of finding a valid \( \text{Solution} \) in response to a query such as

\[
\text{tiling}([\textcolor{red}{[1,1]} \textcolor{blue}{, [1,2]} \ldots, [10,10]],X,[10,10]).
\]

Full marks will only be given for predicates that can exploit backtracking to find all possible solutions. [12 marks]