2003 Paper 13 Question 7

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.

```
tile([[A,B],[C,D],[E,F]]) :- C is A+1, D is B, E is A, F is B-1;
                             C is A+1, D is B, E is A, F is B+1;
                             C is A-1, D is B, E is A, F is B+1;
                             C is A-1, D is B, E is A, F is B-1.
```

Explain what happens in response to a query of the form

```
tile([[4,5],[B,C],[D,E]]).
```

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

(b) Write a Prolog predicate `goodplace([[A,B],[C,D],[E,F]],[N,M])` that is true if `[[A,B],[C,D],[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

```
goodplace([[10,4],[B,C],[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 `tiling(Available,Solution,Size)`. Here, `Size` is the size of the board represented as above, `Solution` is a list of tiles that solves the problem, and `Available` is a list of available positions on a board of the given size. For example, if `Size` is `[2,2]` then `Available` is `[[1,1],[1,2],[2,1],[2,2]]`.

Your predicate should be true if the `Solution` given is a valid one, and should be capable of finding a valid `Solution` in response to a query such as

```
tiling([[1,1],[1,2],...,[10,10]],X,[10,10]).
```

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