

Exercises for the Course

Logic Programming Engineering

(Author Dr. W. Nauber)

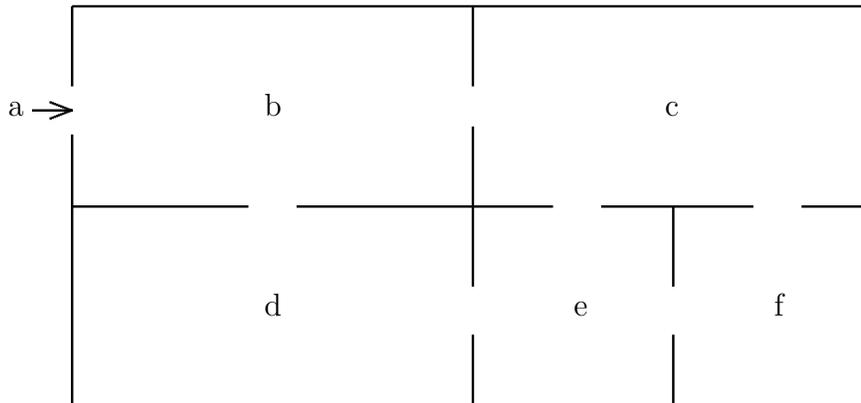
Dr P. Bruscoli, Peter Steinke, Amin Timany

Practical Session 7

Backtracking II, Depth First Search

Exercise 7.1

Given the following maze:



Find a pathway from one given room to another given room, without entering any room twice.

Hints for realization in Prolog:

Define a predicate $maze(Room1, Room2, Way)$ which computes the way from room $Room1$ to room $Room2$, and returns the solution Way as a list of rooms. The existence of a door between two rooms is described as a fact $door(room1, room2)$ (e.g. $door(a, b)$).

Example: $? - maze(a, d, W)$. yields

$W = [a, b, c, f, e, d]$;

$W = [a, b, c, e, d]$;

$W = [a, b, d]$;

false.

Exercise 7.2

From a pack of playing-cards we take 4 cards and place them side by side on a desk. Peter has a glance at the order of cards, and later he remembers the following facts:

1. A knight was laying immediately to the left of a heart card.
2. A queen was laying immediately to the right of a spade card.
3. A diamond card was laying immediately to the right of a king.
4. A club card was not laying immediately next to a heart card but immediately to the left of an ace.

Use a Prolog program to find the original arrangement of the cards.

Exercise 7.3

Six coins are laying on a desk in the following order:

head head head tail tail tail

In one move you can flip two coins which are laying side by side.

(flipping means: head \rightarrow tail, tail \rightarrow head).

Which sequence of moves is needed for the following arrangement of the coins?

tail head tail head tail head

Write a Prolog program which solves this problem.

Hints for realization in Prolog:

Use the search procedure from exercise 7.1 by changing the predicate *maze(Room1, Room2, Way)* to *coins(From, Goal, Way)*. The search starts now with *coins([h, h, h, t, t, t], [t, h, t, h, t, h], Way)* (*h* - heads, *t* - tails). Instead of *way(From, To)* use *move(From, To)* where the predicate *move([X1, X2, X3, X4, X5, X6], [Y1, Y2, Y3, Y4, Y5, Y6])* realizes a move which flips two coins (5 cases). For turning over a coin from *X* to *Y* (*X* and *Y* have values *h* or *t*) define facts for an auxiliary predicate *flipcoin(X, Y)*.