Prolog assignment

November 21, 2021

Choose one of two logic puzzles and implement a solver.

1. You can choose between an easier version, where you are asked to implement a solver for the one particular puzzle described below or to go for a harder version, implements a generalized version (described in the "Generalization" section), which works with arbitrary input. If you choose the former option, you can get at most 10 points from this assignment, and if you choose the latter option, you can get the full 15 points.

2. Two search strategies are expected, chosen from three options: a) depth-first-search, b) iterative-deepening depth-first-search and c) breadth-first-search. Each strategy will give you 50% of the maximum achievable points.

3. You are expected to use only the techniques covered by this course. In particular, implementations using a constraint-satisfaction library (CSP) or definite clause grammar (DCG) will not be accepted.

1 Escape from Zurg

Buzz, Woody, Rex, and Hamm have to escape from Zurg. They merely have to cross one last bridge before they are free. However, the bridge is fragile and can hold at most two of them at the same time. Moreover,

to cross the bridge, a flashlight is needed to avoid traps and broken parts. The problem is that our friends have only one flashlight with one battery that lasts for only 60 minutes. The toys need different times to cross the bridge (in either direction):

Toy	Time
Buzz	5
Woody	10
Rex	20
Hamm	25

Since there can be only two toys on the bridge at the same time, they cannot cross the bridge all at once. Since they need the flashlight to cross the bridge, whenever two have crossed the bridge, somebody has to go back and bring the flashlight to those toys on the other side that still have to cross the bridge. The problem now is: In which order can the four toys cross the bridge in time (that is, within 60 minutes) to be saved from Zurg?

Example solutions:

- `[left_to_right(buzz,woody), right_to_left(buzz), left_to_right(hamm,rex), right_to_left(woody), left_to_right(buzz,woody)];`
- `[left_to_right(buzz,woody), right_to_left(woody), left_to_right(hamm,rex), right_to_left(buzz), left_to_right(buzz,woody)].`

Generalization: The input consists of

- time-limit, which is a positive number T and
- a list of toys and their times to cross the bridge

Example:

```
?- solve(60,[[buzz,5],[woody,10],[rex,20],[hamm,25]], S).
```

2 Verbal arithmetic

Find an assignment of numbers to letters O, M, Y, E, N, D, R and S so that the following equation holds:

$$\begin{array}{cccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

Avoid trivial solutions, where all letters are assigned value 0.

Expected solution:

$$O = 0, M = 1, Y = 2, E = 5, N = 6, D = 7, R = 8 \text{ and } S = 9.$$

Generalization: The input consists of any 3 words in the English alphabet. The first 2 words have an equal length, and the 3rd word is longer by 1 character. We seek a replacement of each letter by a number 0 to 9 so that the first two words are summed to the third one.
Example: `sum([S,E,N,D], [M,O,R,E], [M,O,N,E,Y])`.

Evaluation

Your code has to satisfy several requirements. It

- must have clear instructions as how to run it in SWI Prolog and how to read the results (mandatory),
- should show no compiler warnings when loaded (weight: 10%),
- be legible, well documented and easy to understand (30%),
- implement the specified behaviour without reservations (30%),
- be sound and complete – algorithm finds a solution if and only if the solution exists (30%).