Rattrapage de programmation logique

9 juin 2021

La clarté et la précision de la rédaction seront prises en compte dans l'évaluation.
Le barème est indicatif.

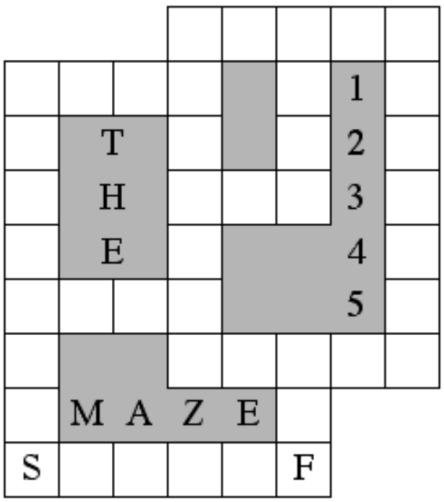
Aucune question ne pourra être posée durant l'examen.
En cas de doute concernant le sujet,
vous poursuivrez votre réponse en expliquant vos hypothèses.
Tous les prédicats que vous définissez doivent être spécifiés.
Choisissez bien vos noms de prédicats et de variables.
`is`, `!` et `=:=` sont INTERDITS.

Pour rendre votre travail par mail, merci d'utiliser comme sujet:
[rattrapage prolog] NOM Prénom groupe

Le nom du fichier pdf rendu sera **GIX_NOM_Prenom_Prolog.pdf** avec X votre numéro de groupe.

Durée : 2h

Dans cet examen, vous allez résoudre le problème "12345 Maze" suivant, initialement proposé par Erich Friedman:
On part de `S`, on se déplace de 1 case dans la direction de votre choix (Nord, Sud, Est, Ouest), et au mouvement suivant, exactement 2 cases (toujours dans la direction de votre choix), puis 3, 4, 5, et on reprend de 1.
Seules les cases blanches sont accessibles, `F` marque l'arrivée.
L'objectif est de trouver la liste des déplacements menant de `S` à `F`.



Modélisation /4pt

Écrire le prédicat `listeModelisation(L)` qui est vrai si la liste `L` est une modélisation du problème, contenant le symbole `b` pour les cases accessibles et le symbole `n` pour les autres.

```
?- listeModelisation(L).
L = [[n,n,n,b,b,b,b,b],[b,b,b,b,n,b,n,b],...]].
```

Transposée /4pt

Écrire le prédicat `listeListeTranspose(M,T)` qui est vrai si la matrice `T` est la transposée de la matrice `M`.

```
?- listeListeTranspose([[1,4],[2,5],[3,6]],T).
T = [[1,2,3],[4,5,6]].
```

Voisinage /4pt

Proposer un prédicat qui définit le voisinage d'une case en fonction de sa position et du nombre de pas ($1, 2, 3, 4$ ou 5). Attention seules les cases `b` sont candidates pour passer d'une case à son voisin.

Chemin /4pt

Proposer un prédicat qui trouve un chemin entre la case `S` et la case `F`, sous forme d'une liste de déplacements.

Iterative Deepening /4pt

Regrouper l'ensemble des prédicats précédents pour trouver la solution au problème en testant les chemins par ordre croissant de longueur. Par exemple en numérotant les lignes `i` de 1 à 9 (de haut en bas) et les colonnes `j` de 1 à 8 (de gauche à droite), on obtient les déplacements `[i,j]` suivants:

```
?- maze(Sol).
Sol = [[9,1],[8, 1], [6, 1], [6, 4], [2, 4], [7, 4], [7, 5], [7, 7], [...|...]|...]
```

♣♢♡♠