Projet Logique

Thomas DUPRIEZ et Guillaume HOCQUET

4 mars 2015

Utilisation

Les programmes sat.cpp et smt.cpp se compilent à l'aide de la commande :

```
g++ -o out sat.cpp
```

Et s'utilisent à l'aide d'un fichier map.txt contenant le niveau :

```
./out map.txt
```

La console indique alors si une solution existe et l'affiche en ASCII le cas échéant. Il est également fournit un fichier solution-0.pdf.

Première partie

Nous avons cherché à relier l'entrée à la sortie à l'aide d'un chemin en faisant en sorte que les diamants soient intégrés à ce chemin.

Pour ce faire, nous avons associé à chaque case quatre variables de direction, une pour chacun de ses cotés (haut, bas, droite et gauche). Ces variables représentent le fait que le chemin solution du niveau traverse le coté associé de la case. Les variables miroirs se voient munis d'une variable supplémentaire correspondant à l'orientation du mirroir.

On impose ensuite des conditions d'équivalence entre les variables de direction des différentes cases :

— Une case *vide* assure la transmission des valeurs de ses variables en adéquation avec ses voisines.
— Une case *mur* empêche toute transmission.
— Une case *miroir* assure une transmission dépendant de son orientation.
— Une case *diamant* requiert un sens de parcourt.
— Les cases *départ* et *arrivée* imposent un sens.

Ainsi, si un chemin existe, il détermine une solution à la formule générée. Réciproquement, une solution permet de retrouver un chemin, qui peut ne pas être contigu dans le cas où il comporterait des cycles.

On assigne à chaque case un numéro en comptant de quatre en quatre de la gauche vers la droite puis de haut en bas en partant de la case en haut à gauche qui reçoit le numéro 1. On calcule les numéros des variables de direction d'une case en partant du numéro de le case et en ajoutant :

- 0 pour le haut
- 1 pour la droite
- 2 pour le bas
- 3 pour la gauche

Deuxième partie

Pour palier aux défauts de la précédente partie, nous allons à présent compter les diamants rencontrés au cours d'un parcours et vérifier à la fin que le compte est bon.

Les variables utilisées sont les suivantes :

— Les variables de direction (booléennes) ont le même usage que dans la partie 1. Leurs noms commencent par c suivis des numéros qu'elles avaient précedemment.

— Les variables d'orientations des miroirs (booléennes) ont le même usage que dans la partie 1. Leurs noms commencent par m et sont suivis des numéros qu'elles avaient précedemment.

— Chaque case posséde maintenant quatres variables (booléennes), une pour chaque direction, exprimants le nombre de diamants rencontrés jusqu'à cette case par un chemin poursuivant dans cette direction. Les noms de ces variables commencent par d et sont suivis des mêmes numéros que les variables de direction de la case.

— Chaque case *diamant* dispose également d'une variable entière qui prendra la valeur 1 si la case en question est traversée deux fois par le chemin et 0 sinon. Les noms de ces variables de double passage commencent par u.

Les variables d transmettent tout au long du chemin le nombre de diamants rencontrés jusqu'à la case courante dans la direction courante. On impose que la case *départ* commence à 0 et que les cases *diamant* incrémentent cette donnée. Pour assurer que les cases *diamant* incrémentent dans le sens de parcours, il est effectivement nécessaire d'avoir recours à 4 variables par case. Enfin, on impose à la case *sortie* de dénombrer tous les diamants disposés sur la map, en comptant éventuellement en double les diamants parcourus 2 fois à l'aide de la somme des variables u.