Projet Logique

Thomas DUPRIEZ et Guillaume HOCQUET

March 4, 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 à fournir :

```
./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 implémenté 4 variables de direction pour chaque case de la grille ainsi que des variables indiquant l'orientation de chaque miroir.

- Une case *vide* assure la transmission dans le même sens en entrée et en sortie.
- Une case *mur* empêche toute transmission.
- Une case *miroir* change la direction en fonction 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.

Les cases sont parcourues de gauche à droite et de haut en bas, elles possèdent donc un numéro en fonction de leur position, auquel on ajoute :

- 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 parcourt et vérifier à la fin que le compte est bon.

Les variables utilisées sont les suivantes :

- Les directions (comme dans la partie 1) commencent par c.
- Les orientations de miroirs (comme dans la partie 1) commencent par m.
- Les nombres de diamants rencontrés commencent par d.
- Les 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. Il y en a donc 4 par case, et 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 parcourt, il est effectivement nécessaire d'avoir recourt 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 des variables u.