

2I008 Procs et Foncs - Projet Robozzle (Partie 2)

Equipe pédagogique 2I008

Année 2014 - 2015

Dans la première partie, vous avez posé les bases de votre version de Robozzle, et votre fonction d'entrée de programme parse un fichier de puzzle, initialise la machine virtuelle (VM) robazzle, puis affiche l'état de cette VM à l'aide du module graphique. Dans cette partie du sujet, on propose d'implanter l'évaluateur pas-à-pas de la machine virtuelle, afin de déplacer le robot selon un programme robazzle donné.

1 Machine virtuelle : fonction d'évaluation

Nous expliquons d'abord le fonctionnement interne de la VM. Celle-ci interprète du code compilé (bytecode) qui est le résultat de la compilation d'un programme robazzle.

Le champ `code` de la VM contient un tableau dans lequel les fonctions F1 à F5 compilées en bytecode sont stockées à la suite. Afin de ne pas exécuter une fonction robazzle à la suite de la précédente sans qu'elle n'ait été appelée, on insérera une instruction `Exit` à la fin de chaque fonction. Les adresses pour les sauts (instructions `Jump` et `JumpIfNot`) seront calculées à la compilation, vous pouvez les compter à la main sur de petits programmes pour vos tests. La VM utilise également plusieurs registres :

- `pc` est le pointeur de code, il pointe sur la prochaine instruction à exécuter dans le tableau de bytecode,
- `star` maintient le nombre total d'étoiles encore présentes dans la map,
- `pos` et `dir` maintiennent la position et la direction courante du robot,

ainsi qu'une pile (le champ `stack`). Cette pile est utilisée lors d'appels (récursifs ou non) aux fonctions robazzle F1 à F5. Lors d'un appel à une fonction, on dépose l'adresse de retour (`pc + 1`) sur la pile. Lorsque l'on rencontre l'instruction `Return`, l'adresse de retour est dépilée, l'exécution reprendra ensuite à cette adresse.

Question 1.1. Implémenter la fonction `set_code : state -> bc array -> state` du module `Vm`. Cette fonction retourne un nouvel état de machine virtuelle dans lequel le segment de code a été renseigné.

Question 1.2. Implémenter la fonction `step : state -> state` du module `Vm`. Cette fonction récupère l'instruction courante (la `pc`-ième du tableau de bytecode) et l'interprète.

2 Boucle d'évaluation

Le module de la machine virtuelle `Vm` vous fournit maintenant une fonction d'évaluation pas-à-pas `step`. Il s'agit maintenant d'implanter la boucle d'évaluation d'un programme bytecode complet, dans le fichier `main.ml`, afin de tester votre interprète de bytecode.

Question 2.1. Implémenter la fonction récursive terminale `run : Vm.state -> Vm.state`, qui calcule le prochain état de la VM passée en paramètre. Les conditions de sorties sont : niveau terminé, plus de bytecode à exécuter, ou robot en dehors de la map. Vous utiliserez notamment la fonction `G.clear` qui efface l'écran, ainsi que la fonction d'affichage de la VM que vous avez implémentée dans la partie 1 du sujet.

Question 2.2. Modifier la fonction principale du fichier `main.ml`, en y ajoutant la déclaration d'un programme bytecode (de type `Vm.bc array`), et l'évaluation de ce programme.

Vous devriez maintenant voir le robot se déplacer, et peut-être même ramasser toutes les étoiles si votre solution du puzzle est correcte !