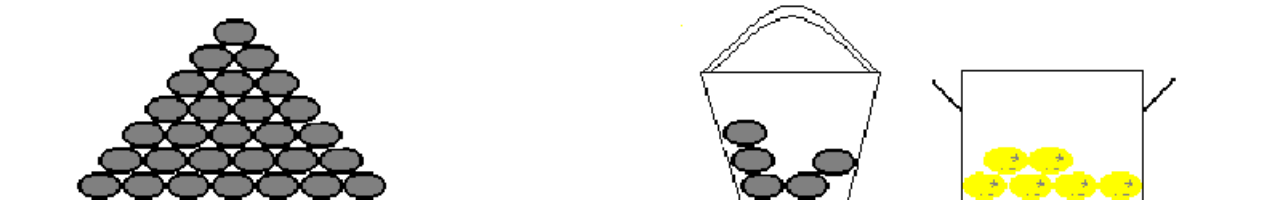


Robot peleur

1 Description du problème

On dispose d'un robot pour exécuter une tâche de pelage de pomme de terre. Cet exécutant évolue dans un environnement schématisé comme suit :



Il dispose d'un panier qu'il peut aller remplir à un tas de pommes de terre. Ce tas est de taille suffisante pour satisfaire n'importe quelle commande : nous dirons qu'il est inépuisable. Le robot est aussi capable de prendre une pomme de terre dans le panier, de la peler et de la jeter dans la marmite.

2 Description formelle du robot

Les conditions de travail de notre robot étant précisées, il reste à le caractériser formellement en tant qu'exécutant : il est indispensable pour cela, d'indiquer quelles instructions d'action élémentaire vont le faire agir et quelles sont les conditions qu'il est capable de tester.

A Instructions d'action élémentaire

On peut en dresser la liste sous la forme d'un lexique :

- ▶ `Remplis` : il va remplir le panier et le ramène près de son lieu de travail ;
- ▶ `Pèle` : il prend une pomme de terre dans le panier, la pèle et la place dans la marmite.

Ce lexique traduit un ordre formel¹ (`Remplis`) compris par le robot en une explication, pour le programmeur, de l'action qui sera réalisée par le robot lorsque cet ordre sera invoqué.

Il s'agit là des deux seules instructions que le robot « comprend » et est capable d'exécuter. Il est donc interdit de faire figurer d'autres instructions que celles-là dans la marche à suivre.

B Les conditions que l'exécutant peut tester

On peut en dresser la liste des condition sous la forme d'un lexique :

- ▶ `la marmite est pleine` : condition *vraie* lorsque la marmite est pleine !
- ▶ `le panier est vide` : condition *vraie* lorsque le panier est vide !

¹ Notez le changement de typographie auquel je m'astreins : les caractères en police `Courier` indiquent qu'il s'agit d'instructions, de structures de contrôle ou de conditions directement compréhensibles par le robot.

Ces conditions peuvent figurer au sein des structures alternatives et répétitives. De plus, il est permis de former de nouvelles conditions plus complexes à partir des conditions précédentes :

- ▶ en énonçant le contraire d'une condition (*e.g.* le panier n'est pas vide) ;
- ▶ en liant les conditions par des opérateurs logiques « ET » et « OU » (*e.g.* le panier est vide ET la marmite est remplie).

C Précisions nécessaires

Pour finaliser l'étape de « quoi faire ? » menant à un cahier des charges complet, c'est-à-dire ne laissant aucun espace à interprétation pour l'étape du « comment faire faire ? ». Tout un tas de **spécifications** complémentaires du problème sont nécessaires. Elles portent sur l'état initial du robot, sur l'état final attendus et également des précisions sur les instructions.

- ▶ Vous ne pouvez rien supposer quant à l'état initial du panier lorsque le robot commencera l'exécution de votre marche à suivre : il est peut-être vide, peut-être plein, peut-être à moitié vide ...
- ▶ De plus, vous ne pouvez rien conjecturer non plus à propos de sa taille : il peut être tout petit (ne pouvant contenir qu'une pomme de terre) ou très grand !
- ▶ Vous ne pouvez non plus présumer de la taille de la marmite à remplir, ni de la quantité de pommes de terre déjà pelées qui y est contenue. La marche à suivre doit être valable, qu'il s'agisse d'une énorme marmite, d'une marmite ne pouvant contenir qu'une seule pomme de terre ou même d'une marmite qu'on amène déjà à moitié ou complètement remplie (dans ce dernier cas, le robot ne doit évidemment rien peler !).

Ces informations peuvent vous apparaître comme des contraintes supplémentaires du problème. Mais ce sont, bien au contraire, des aides précieuses qui résultent de l'étape la plus difficile de la programmation : le « quoi faire ? » ! Se poser les questions relatives à ces contraintes vous mènera tout droit vers la solution algorithmique.

3 But de l'exercice

Construire l'algorithme qui permettra au robot d'agir tel que les spécifications précédentes l'ont décrit