Faculté des Sciences
Faculté des Sciences Appliquées

Conception fiable de systèmes distribués de contrôle d'équipement avec les Robot LegoMindstorm

Introduction

Qu'est-ce que la vérification ?

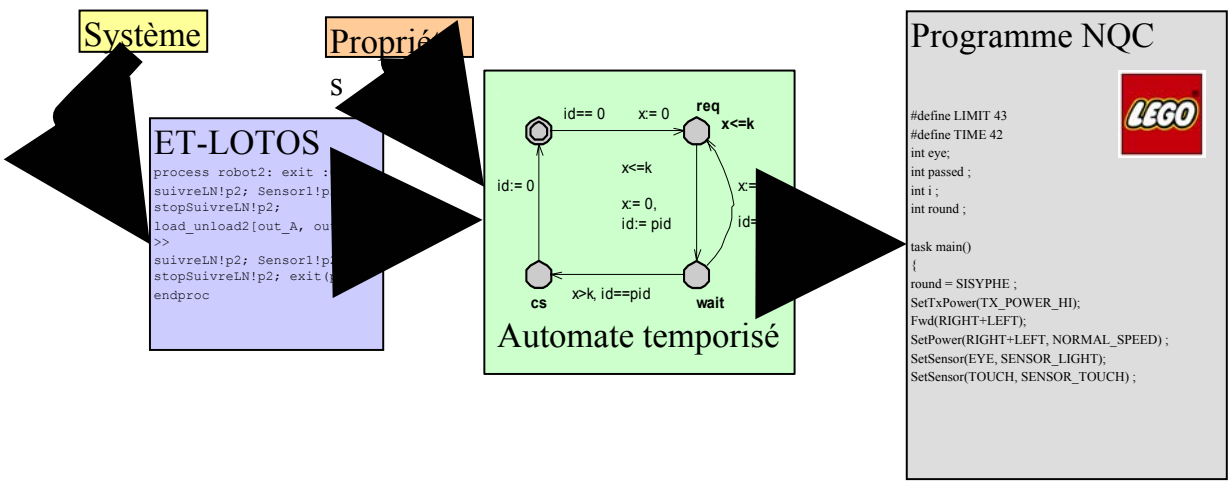
Écrire un programme n'est pas une chose simple ! Même en y apportant beaucoup de soin, il peut arriver que des erreurs subsistent. Or, il existe des contextes dans lesquels l'erreur n'est pas tolérable: de nos jours, les programmes informatiques contrôlent les fusées, les métros, les centrales nucléaires.

La « vérification » permet de garantir le bon fonctionnement de ce type de programme.

Comment vérifier ?

On ne peut pas, en l'état actuel des connaissances, vérifer directement un programme donné. C'est pourquoi nous devons simplifier le problème. Tout comme la physique fait appel à des modèles pour expliquer et représenter les phénomènes réels (le célèbre E=mc², par exemple), nous aurons recours à un modèle du programme. Nous appelons ce modèle une abstraction, ou spécification formelle.

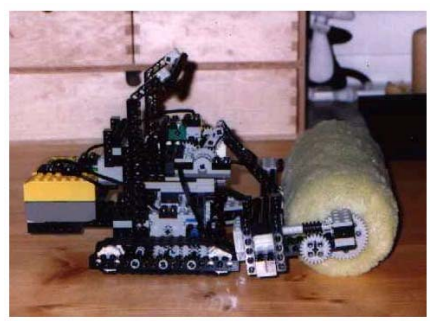
Le processus de vérification



La propriété qui nous intéresse: l'exclusion mutuelle

Pour empêcher toute catastrophe qui serait fatale à nos deux robots, nous voulons à tout prix éviter qu'ils se présentent ensemble au lieu de déchargement. Cette propriété est un cas d'exclusion mutuelle.

On assure cette propriété en utilisant les possibilités de communication de la brique LEGO®™©. Une fois qu'un des deux robots a déposé sa brique, il envoie à l'autre un message pour lui passer la main. Au départ, le robot Sisyphe possède le tour.



On voit tout de suite que si un des deux robots tombait en panne, tout le système serait bloqué (car l'autre attendrait indéfiniment de recevoir le tour !) Pour résoudre ce problème, on permet à un robot de prendre le tour s'il n'a pas reçu de nouvelles de l'autre au bout d'un certain délai (*timeout*). Celui qui prend ainsi le tour envoie alors un message à l'autre pour lui dire qu'il est maintenant « hors jeu »