TD 3 : Modélisation à l'aide de machines à états finis

Yann Régis-Gianas (yrg@pps.jussieu.fr)
GL6 - Licence 3 - Université Paris 7

7 avril 2009

Au terme de cette séance, vous devrez savoir :
- ✓ déterminer les états utiles à la modélisation d'un problème ;
- ✓ énumérer les événements/actions représentant l'interaction d'un système avec son environnement ;
- ✓ calculer les interactions potentielles entre plusieurs systèmes de transitions ;
- ✓ vous être initiés aux problèmes de synchronisation entre plusieurs systèmes ;
- ✓ être capable de vérifier certaines propriétés d'un système de transition.

1 Automate modélisant une pile bornée

Un **système de transitions étiquetées** est un triplet $(Q, \Lambda, \rightarrow)$ où Q est un ensemble (potentiellement infini) d'états, Λ est un ensemble (potentiellement infini) d'étiquettes et $\rightarrow$ est une relation ternaire inclue dans $Q \times \Lambda \times Q$. On utilise la relation $p \xrightarrow{\alpha} q$ lorsque trois objets $p \in Q$, $\alpha \in \Lambda$ et $q \in Q$ sont en relation par $\rightarrow$.

Un **automate fini déterministe** est un quintuplet $(Q, \Sigma, \delta, I, F)$ où Q est un ensemble fini d'états, Σ est un alphabet, δ une application de $Q \times \Sigma$ dans Q, I est l'ensemble des états initiaux, et F l'ensemble des états finals.

Pour modéliser un système à l'aide d'un automate fini ou d'un système de transition, il est donc nécessaire de répondre à ces questions :

1. Quels sont les différents états du système ? Est-ce que l'ensemble des états est fini ?
2. À quels événements réagit ce système ?
3. Quelles transformations sur l'état du système sont induites par ces événements ? Est-ce que ces transformations dépendent uniquement de l'entrée et de l'état courant ? Est-ce que le système est déterministe ?

Une modélisation sert avant tout **à comprendre et analyser** un système (existant ou en cours de conception). La façon dont les états du système sont représentés peut donc simplifier certains aspects du système en prenant un point de vue particulier.

❶ À l'aide d'un automate fini, spécifiez les transformations possibles d'une pile bornée pouvant contenir au plus deux éléments de l'ensemble $\{a, b\}$ et munie des opérations $\textsc{Push}$, $\textsc{Pop}$ et $\textsc{Top}$ définies dans votre projet.

❷ Que représente le langage reconnu par cet automate ?

❸ Comment déduire les évolutions possibles de la hauteur de la pile à partir de la spécification précédente ?

❹ Comment utiliser cette spécification sous forme d'automate pour prouver que l'implémentation de pile bornée de votre projet (à partir d'un tableau et d'un entier) est correcte ?

❺ Quelle spécification donner à une pile bornée pouvant contenir au plus trois éléments ? au plus N éléments ? Quelle spécification donner à une pile non bornée ?

❻ Est-ce que la spécification de l'opération $\textsc{Top}$ est satisfaisante ? Pourquoi ?

Un **automate de Mealy** est un sextuplet $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ où Q est un ensemble fini d'états, q_0 est un état initial, Σ est l'alphabet d'entrée, Δ est l'alphabet de sortie, δ est une application de $Q \times \Sigma$ dans Q et λ est une application de $Q \times \Sigma$ dans Δ.

Un **automate de Moore** est un sextuplet $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ où Q est un ensemble fini d'états, q_0 est un état initial, Σ est l'alphabet d'entrée, Δ est l'alphabet de sortie, δ est une application de $Q \times \Sigma$ dans Q et λ est une application de Q dans Δ.

❼ Exprimez de nouveau la spécification de la première question à l'aide d'automates de Mealy et de Moore.

2 Automate modélisant une interface

Un éditeur de texte fournit une interface d'interaction que l'on décrit informellement de la façon suivante :
- si l'utilisateur utilise l'action **ouvrir** f (où f est un nom de fichier) :
  - un tampon de travail est créé à partir du contenu de f ;
  - si un tampon de travail existe déjà, il est remplacé par un nouveau tampon défini par le contenu de f.
- si l'utilisateur utilise l'action **fermer**, le tampon courant est détruit.
- si l'utilisateur utilise l'action **éditer**, le contenu du tampon de travail est modifié.
- si l'utilisateur utilise l'action **sauvegarder**, le contenu du fichier associé au tampon courant est remplacé par le contenu de ce tampon.

❶ Énumérez les différentes composantes de l'état de l'application. Est-ce que toutes les combinaisons d'états sont valides ?

❷ Modélisez le comportement de l'application à l'aide d'un automate fini.

❸ Est-ce que cette spécification garantit la propriété suivante ?

Lorsque l'application se termine, l'état du tampon et du fichier qui lui est associé sont identiques.

❹ Modifiez la spécification pour résoudre le problème de la question précédente (indice : augmentez les interactions possibles entre l'utilisateur et l'application).

3 Synchronisation des composants d'une photocopieuse

Une photocopieuse est composée de deux modules :
- une interface utilisateur composée d'un unique bouton. Si on l'actionne une copie est mise en route. Si une copie est déjà en cours, alors actionner le bouton n'a aucun effet visible.
- un copieur qui est en attente d'une commande si il n'est pas déjà en fonctionnement.

❶ Énumérez les états des deux modules ainsi que les événements qui affectent leurs comportements.

❷ Associez à chaque module une machine à états finis spécifiant son comportement.

❸ Comment faire interagir ces machines ?

❹ Construisez un automate modélisant cette interaction.

❺ Qu'arrive-t-il si on appuie deux fois (très rapidement) sur le bouton de commande ?

❻ Modélisez une photocopieuse permettant le stockage d'une file de taille N de travaux à effectuer successivement.

4 Compteur binaire

❶ Modélisez le comportement d'un *bit* vis-à-vis de deux messages **raz** et **set**.

❷ Modélisez l'interaction entre un *bit* de poids fort et un *bit* de poids faible.

❸ Généralisez ce procédé pour construire un mot de N *bits*.

❹ On suppose maintenant qu'un *bit* puisse être défectueux en prenant en compte un nouveau message **error**. Quelles conséquences impliquent ce nouveau comportement sur la modélisation de l'interaction entre les *bits* ?