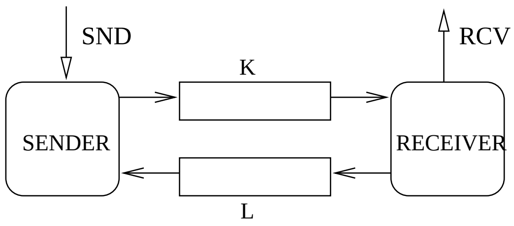
Devoir :
Modélisation et vérification avec SPIN du protocole du bit alternant

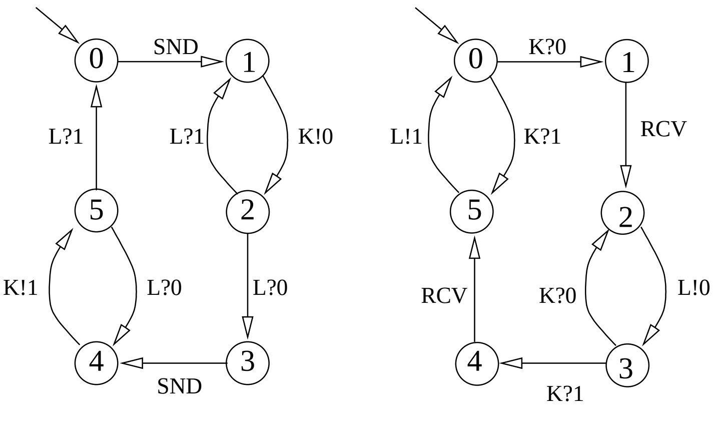
A rendre par e-mail à `Mihaela.Sighireanu@liafa.jussieu.fr` au plus tard le 18 janvier 2001

Ce document est disponible à l'URL
`http://www.liafa.jussieu.fr/~sighirea/logver/cc_spin_altbit.html`

L'architecture générale du protocole du bit alterné est :



et les automates qui décrivent le fonctionnement de l'émetteur (automate (a)) et du récepteur (automate (b)) sont :



(a) (b)

On désire modéliser ce protocole en Promela et puis vérifier des formules Ltl en utilisant Spin.

1. On modélise les canaux de communication K et L par des canaux Promela ayant une longueur constante (à choisir), strictement positive. Le type des messages qui circulent sur ces canaux est `bit`, c'est-à-dire qu'on fait abstraction des messages envoyés et on considère uniquement le bit qui accompagne les messages. Donner la déclaration des canaux de communication K et L.

2. On modélise les canaux de communication SND et RCV par deux variables de type `bit`, c'est-à-dire qu'on fait abstraction des messages envoyés par les clients et on suppose qu'il sont égaux au bit alterné émis resp. reçu. Donner la déclaration de SND et RCV.

3. On modélise l'émetteur (processus SENDER) par un processus PROMELA qui implémente l'automate (a). Ce processus est paramétré par deux canaux : `in` sur lequel l'émetteur reçoit des messages et `out` sur lequel l'émetteur envoie des messages. Pour la modélisation de l'automate, utiliser les étiquettes et les instructions "`goto`" du C. Pour la modélisation la transition étiquetée par SND, affecter la variable `SND` au bit alterné émis et afficher un message (avec "`printf`"). Donner la définition du processus SENDER en utilisant un nombre minimum d'états (< 6).

4. On modélise le récepteur (processus RECEIVER) par un processus PROMELA qui implémente l'automate (b). Ce processus est paramétré par deux canaux : `in` sur lequel le récepteur reçoit des messages et `out` sur lequel le récepteur envoie des messages. Pour la modélisation la transition étiquetée par RCV, affecter la variable `RCV` au bit alterné reçu et afficher un message (avec "`printf`"). Donner la définition du processus RECEIVER en utilisant un nombre minimum d'états (< 6).

5. Donner la définition du processus "`init`".

6. Simuler avec SPIN le comportement du système ainsi modélisé.

7. Donner la formula LTL qui exprime la propriété p : "tout message envoyé par l'émetteur est reçu par le récepteur sans erreur au moins une fois".

8. Vérifier avec SPIN cette propriété et expliquer (en quelques lignes) le résultat obtenu.

9. Modifier la définition des processus SENDER et RECEIVER pour modéliser l'occurrence d'une erreur à la réception de l'acquittement resp. du message. Il s'agit d'ajouter à l'automate (a) deux transitions étiquetées `err`, l'une partant de l'état 2 vers l'état 1 et l'autre partant de l'état 5 vers l'état 4. Pareil pour l'automate (b) : de l'état 3 vers l'état 2 et de l'état 0 vers l'état 5. Pour la modélisation, utiliser "`printf`" sur signaler l'occurrence d'une erreur.

10. Pour cette nouvelle modélisation, vérifier la propriété p avec SPIN et expliquer (en quelques lignes) le résultat obtenu.