

---

Mounir Lallali, Prof. Burkhart Wolff  
Parc Orsay Université  
4, rue Jacques Monod  
Building H / Room 012

## TD 2

Date : 29.09.2009, Durée : 3 heures

### Exercice 1

1. Donner l'automate associé à la spécification LOTOS (de base) suivante :

$$DLC[in, out] := in ; (out ; DLC[in, out] \square DLC[in, out])$$

2. Soit maintenant les processus :

$$IN[in] := in ; IN[in]$$

$$INOOUT[in, out] := in ; out ; INOOUT[in, out]$$

Quels sont les comportements de :

$$DLC[in, out] \parallel IN[in]$$

$$DLC[in, out] \parallel INOOUT[in, out]$$

3. On considère maintenant

$$NDLC[in, out] := in ; (out ; NDLC[in, out] \square i ; NDLC[in, out])$$

Quels sont les comportements de :

$$NDLC[in, out] \parallel IN[in]$$

$$NDLC[in, out] \parallel INOOUT[in, out]$$

### Exercice 2 (Protocole du bit alterné)

**Description du service** Le protocole du bit alterné (alternating bit protocol) fait partie de la couche (4ème couche du modèle OSI). Il permet le transfert de données entre une paire d'entités pour lesquelles une connexion bidirectionnelle a été préalablement établie. Pour simplifier le problème, on crée une dissymétrie entre les deux entités : la première (T, comme **t**ransmitter) émet des messages à destination de la seconde (R, comme **r**eceiver). Les messages sont modélisés par des numéros compris entre 1 et un entier maximal  $N$  ; ils sont spécifiés par le type abstrait suivant, qui ne précise pas leur nature exacte :

```

type MESSAGE is
  sorts MSG (* type MSG = 1..N *)
endtype

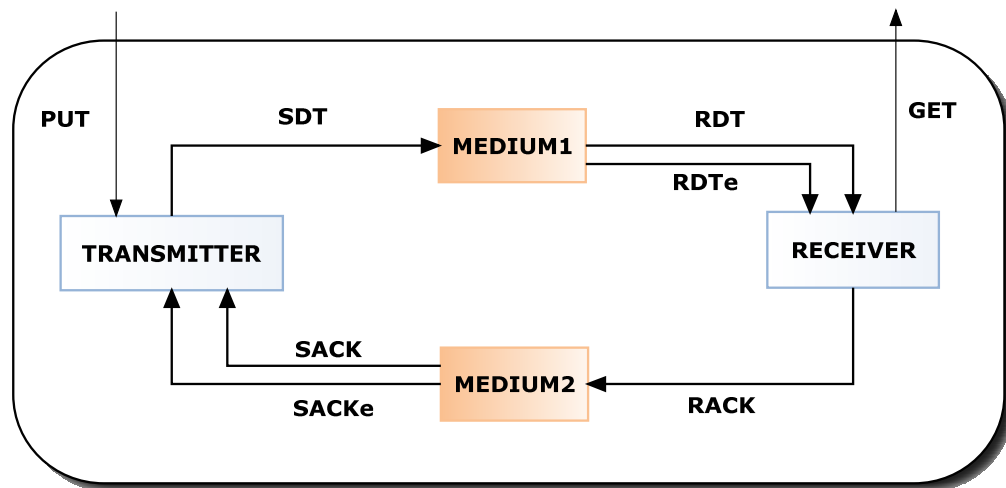
```

Vu de la couche supérieure, le service fourni par le protocole du bit alterné est l'acheminement d'une série de messages de T vers R. La transmission est fiable : les messages ne peuvent pas être perdus ni dupliqués et ils sont reçus dans l'ordre où ils sont émis.

**Question 1 :** Donner la spécification en LOTOS de ce comportement.

**Description du protocole** Le fonctionnement "idéal" du protocole du bit alterné est le suivant : T envoie un message à R ; à la réception de ce message, R renvoie un acquittement à T. La liaison entre T et R n'est pas fiable : Il est possible que des messages ou des acquittements soient perdus. En cas de perte, le medium peut, de manière facultative, signaler cette perte au destinataire (T ou R) en envoyant une indication de perte. Pour détecter les pertes non signalées, les messages et les acquittements contiennent un bit de contrôle. Le bit de contrôle de deux messages successivement émis ont des valeurs distinctes (la valeur du bit alterne à chaque émission). Si l'entité T reçoit une indication de perte d'acquittement ou un acquittement avec un bit de contrôle erroné, elle réémet le dernier message envoyé. Si l'entité T reçoit une indication de perte de message ou un message avec un bit de contrôle erroné, elle réémet le dernier acquittement envoyé.

**Architecture du protocole** On choisit de décrire le protocole par quatre processus parallèles communicants :



Process	Signification
TRANSMITTER	entité émettrice T
RECEIVER	entité réceptrice R
MEDIUM1	Transmission des messages de T vers R
MEDIUM2	Transmission des acquittements de R vers T

Le tableau suivant donne la liste des signaux utilisés. Les seuls signaux fournis par le service sont PUT et GET ; tous les autres désignent des signaux internes. Dans ce tableau, M désigne un message et B désigne le bit de contrôle.

Signal	Origine	Destination	Signification
PUT !M	service	TRANSMITTER	émission du message
SDT !M !B	TRANSMITTER	MEDIUM1	envoi du message
RDT !M !B	MEDIUM1	RECEIVER	transmission du message
RDTe	MEDIUM1	RECEIVER	perte du message
GET !M	RECEIVER	service	réception du message
RACK !B	RECEIVER	MEDIUM2	renvoi d'un acquittement
SACK !B	MEDIUM2	TRANSMITTER	transmission de l'acquittement
SACKe	MEDIUM2	TRANSMITTER	perte de l'acquittement

Dans la suite, on utilisera le type :

```

type BIT is
  sorts BIT
      opns
      0 : BIT
      1 : BIT
      not : BIT BIT
endtype

```

**Question 2 :** Spécifier l'architecture du protocole sans expliciter les définitions des processus TRANSMITTER, RECEIVER, MEDIUM1 et MEDIUM2. On remarquera que l'émetteur et le récepteur fonctionnent en parallèle de manière asynchrone. Il en est de même pour les deux média.

**Spécification du medium des messages** En recevant un message M avec un bit de contrôle égal à B, le medium 1 peut réagir de trois façons différentes :

- Transmettre correctement le message et son bit de contrôle. En aucun cas le medium ne peut changer la valeur du bit de contrôle
- Perdre le message et envoyer une indication de perte à l'entité réceptrice
- Perdre silencieusement le message

**Question 3 :** Écrire la spécification du processus MEDIUM1.

**Spécification du medium des acquittements** Le fonctionnement du medium 2 est analogue à celui du medium 1. La seule différence réside dans les noms de signaux et dans le fait que les acquittements, contrairement aux messages, ne portent pas d'information autre que le bit de contrôle.

**Question 4 :** Écrire la spécification du processus MEDIUM2.

**Spécification de l'émetteur** L'entité émettrice acquiert un message via **PUT** et le transfère au médium 1 après lui avoir ajouté la valeur courante **B** du bit de contrôle. Si elle reçoit en réponse un acquittement avec un bit de contrôle **B**, la transmission a réussi, sinon il faut réémettre le message. Il y a 3 causes possibles de réémission :

- L'émetteur a reçu un acquittement ayant  $(\neg B)$  comme bit de contrôle
- L'émetteur a reçu une indication de perte d'acquittement **SACKe**
- L'émetteur peut réémettre spontanément le message afin d'éviter le blocage dans le cas où le médium 1 (resp. 2) aurait perdu silencieusement un message (resp. un acquittement). En réalité, cette réémission n'a lieu que si une certaine contrainte de délai (timeout) est vérifiée, mais LOTOS ne permettant pas d'exprimer le délai, on le modélise par un événement silencieux "i".

**Question 5 :** Écrire la spécification du processus **TRANSMITTER**. Il peut être utile d'introduire un processus auxiliaire.

**Spécification du récepteur** Lorsqu'elle reçoit un message avec un bit de contrôle **B** correct, l'entité réceptrice délivre le message via **GET** et renvoie un acquittement avec un bit de contrôle égal à **B**. Dans les autres cas, elle renvoie un acquittement incorrect (ayant  $(\neg B)$ ) comme bit de contrôle); ces cas sont au nombre de trois :

- Le récepteur a reçu un message ayant  $(\neg B)$  comme bit de contrôle
- Le récepteur a reçu une indication de perte de message **RDTe**
- Le récepteur peut émettre spontanément un acquittement invalide afin d'éviter le blocage dans le cas où le médium 1 (resp. 2) aurait perdu silencieusement un message (resp. un acquittement). Comme expliqué plus haut, il s'agit d'un timeout que l'on modélise par un événement silencieux "i".

**Question 6 :** Écrire la spécification du processus **RECEIVER**.