

Cours Algorithmique des systèmes parallèles et distribués
 Exercices
 Série sur les protocoles de communication
 par Dominique Méry
 5 février 2026

Exercice 1 (*disapp_td2_ex1.tla*, *disapp_td2_ex1_pluscal.tla*)

Modéliser en TLA⁺ l'envoi d'un message m à un processus P_2 via un canal $CHAN$ par P_1

Exercice 2 (*disapp_td2_ex2.tla*)

Trois processus P_1 , P_2 et P_3 réalisent les actions suivantes :

disapp

- P_1 calcule la fonction f_1 en appliquant cette fonction sur les valeurs se trouvant sur un tas T .
- P_2 calcule la somme des valeurs produites par le processus P_1 .
- P_3 produit les valeurs utilisées par P_1 .

Modéliser ce système en TLA⁺.

Exercice 3 (*disapp_td2_ex3.tla*)

On peut définir un algorithme réparti comme un ensemble d'algorithmes locaux et on définit les systèmes de transition associées comme suit.

Given a set \mathcal{LC} of configurations a set $\mathcal{LI} \subseteq \mathcal{LC}$ of initial configurations, and a set \mathcal{M} of messages, a local algorithm \mathcal{LA} is a structure $(\mathcal{LC}, \mathcal{LI},$

$\rightarrow_i, \rightarrow_s, \rightarrow_r, \mathcal{M}$) with :

- $\rightarrow_i \subseteq \mathcal{LC} \times \mathcal{LC}$ modelling internal computation steps,
- $\rightarrow_s \subseteq \mathcal{LC} \times \mathcal{M} \times \mathcal{LC}$ modelling sending steps,
- $\rightarrow_r \subseteq \mathcal{LC} \times \mathcal{M} \times \mathcal{LC}$ modelling receiving steps.

A distributed algorithm for a collection of processes is a collection $\{\mathcal{LA}_1, \dots, \mathcal{LA}_n\}$ of local algorithms, one algorithm $\mathcal{LA}_k = (\mathcal{LC}_k, \mathcal{LI}_k, \rightarrow_i^k, \rightarrow_s^k, \rightarrow_r^k, \mathcal{M})$ for each process P_k , with a transition relation \rightarrow defined over the set $\mathcal{C} = \mathcal{LC}_1 \times \dots \times \mathcal{LC}_n \times (\mathcal{M} \rightarrow \mathbb{N})$ of configurations : let $C = (C_1, \dots, C_n, M)$ and $C' = (C'_1, \dots, C'_n, M')$ two configurations and let define $C \rightarrow C'$:

- internal transition $\exists k \in \{1, \dots, n\} : (\forall j \in 1..n : j \neq k : C_j = C'_j) \wedge C_k \rightarrow_i^k C'_k \wedge M' = M$
- send transition $\exists k \in \{1, \dots, n\} : \exists m \in \mathcal{M} : \begin{cases} \forall j \in 1..n : j \neq k : C_j = C'_j \\ \wedge \forall o \in \mathcal{M} \setminus \{m\} : M'(o) = M(o) \\ \wedge M'(m) = M(m)+1 \wedge (C_k, m, C'_k) \in \rightarrow_s^k \end{cases}$
- receive transition $\exists k \in \{1, \dots, n\} : \exists m \in \mathcal{M} : M(m) \neq 0 : \begin{cases} \forall j \in 1..N : j \neq k : C_j = C'_j \\ \wedge \forall o \in \mathcal{M} \setminus \{m\} : M'(o) = M(o) \\ \wedge M(m) = M'(m)+1 \wedge (C_k, m, C'_k) \in \rightarrow_r^k \end{cases}$

Ecrire un module TLA⁺ qui décrit les algorithmes locaux constituant un algorithme réparti et modéliser l'algorithme réparti lui-même. Traduire la modélisation des algorithmes locaux et répartis dans la notation TLA⁺.

Exercice 4 (*distapp_td2_ex4.tla*)

*Nous considérons les protocoles de communication selon diverses hypothèses.
Ecrire une solution pour la communication FIFO en intégrant les différents cas d'erreurs ou non.*

Exercice 5 (*a_abp.tla,a_pluscal_abp.tla*)

L'algorithme du bit alterné permet de contrôler la perte possible de messages en proposant un mécanisme basé sur un accusé de réception. Ecrire une solution pour l'algorithme du bit alterné

— *en utilisant le langage TLA⁺.*

— *en utilisant d'abord le langage PlusCal, puis en traduisant en TLA⁺*

Dans les deux cas, il est demandé de vérifier les propriétés attendues.