Cours Algorithmique des systèmes parallèles et distribués
Exercices
Série 2 : Protocoles de communication
par Dominique Méry
27 janvier 2021

Exercice 1

Question 1.1 *Modéliser en TLA$^+$ l'envoi d'un message m à un processus P via un canal CHAN par Q*

Question 1.2 *Modéliser en TLA$^+$ le broadcast d'un message à tous les processus P en lien avec Q*

Exercice 2

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

— *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 *(local and distributed algorithms)*
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 $(\mathcal{LC}, \mathcal{LI}, \longrightarrow_i, \longrightarrow_s, \longrightarrow_r, \mathcal{M})$ with :

— *$\longrightarrow_i \subseteq \mathcal{LC} \times \mathcal{LC}$ modelling internal computation steps,*
— *$\longrightarrow_s \subseteq \mathcal{LC} \times \mathcal{M} \times \mathcal{LC}$ modelling sending steps,*
— *$\longrightarrow_r \subseteq \mathcal{LC} \times \mathcal{M} \times \mathcal{LC}$ modelling receiving steps.*

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

— *internal transition $\exists k \in \{1, \ldots, n\} : \left(\forall j \in 1..n : j \neq k : C_j = C_j' \right) \wedge C_k \longrightarrow_i^k C_k' \ \wedge \ M' = M$*
— *send transition $\exists k \in \{1, \ldots, n\} : \exists m \in \mathcal{M} : \begin{cases} \forall j \in 1..n : j \neq k : C_j = C_j' \\ \wedge \forall o \in \mathcal{M} \backslash \{m\} : M'(o) = M(o) \\ \wedge M'(m) = M(m)+1 \wedge (C_k, m, C_k') \in \ \longrightarrow_s^k \end{cases}$*

— *receive transition* $\exists k \in \{1, \ldots, n\} : \exists m \in \mathcal{M} : M(m) \neq 0 : \left\{ \begin{array}{l} \forall j \in 1..N : j \neq k : C_j = C'_j \\ \wedge \forall o \in \mathcal{M} \backslash \{m\} : M'(o) = M(o) \\ \wedge M(m) = M'(m)+1 \wedge (C_k, m, C'_k) \in \ \longrightarrow^k_r \end{array} \right.$

Ecrire un module TLA$^+$ qui décrit les algorithmes locaux constituant un algorithme réparti et modéliser l'algorithme réparti lui-même.

Exercice 4

Traduire la modélisation des algorithmes locaux et répartis dans la notation TLA$^+$.

Exercice 5

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 6

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 7 **Question 7.1** *Ecrire une solution pour l'algorithme du bit alterné.*

Question 7.2 *Le protocole appelé Sliding Window Protocol est fondé sur un fenêtre qui glisse pour valider progressivement les envois reçus. Le protocole est donné sous la forme d'invariant avec des événements. Proposer un schéma de traduction pour cet algorithme réparti en un module TLA+.*

EVENT e ***ANY*** x ***WHERE*** $G(x, u)$ ***THEN*** $u := f(x, u)$ ***END***	$\rightsquigarrow$	$BA(e) \stackrel{def}{=} \exists x. \begin{pmatrix} \wedge\ G(x, u) \\ \wedge\ u' = f(x, u) \end{pmatrix}$

Question 7.3 *Proposer un schéma de traduction pour un algorithme réparti en TLA$^+$.*

Question 7.4 *Reprendre la solution précédente pour modéliser chan comme un buffer de taille la taille de la fenêtre.*

AXIOMS

$axm1 : n \in \mathbb{N}_1$

$axm2 : IN \in \mathbb{N} \nrightarrow D$

$axm3 : dom(IN) = 0 \mathrel{..} n$

$axm4 : l \in \mathbb{N}$

$axm5 : l \leq n$

VARIABLES OUT, i, ack, got, b

INVARIANTS

$inv1 : OUT \subseteq IN$

$inv2 : 0 \mathrel{..} i{-}1 \lhd OUT = 0 \mathrel{..} i{-}1 \lhd IN$

$inv3 : i \in 0 \mathrel{..} n{+}1$

$inv4 : ack \cup got \subseteq i \mathrel{..} i{+}l \cap 0 \mathrel{..} n$

$inv5 : ack \subseteq dom(OUT)$

$inv1 : OUT \in \mathbb{N} \nrightarrow D$

$inv2 : i \in 0 \mathrel{..} n{+}1$

$inv3 : 0 \mathrel{..} i{-}1 \subseteq dom(OUT) \wedge dom(OUT) \subseteq 0 \mathrel{..} n$

$inv8 : ack \subseteq \mathbb{N}$

$inv10 : got \subseteq \mathbb{N}$

$inv13 : got \subseteq dom(OUT)$

$inv14 : ack \subseteq dom(OUT)$

$inv16 : 0 \mathrel{..} i{-}1 \lhd OUT = 0 \mathrel{..} i{-}1 \lhd IN$

EVENT INITIALISATION

BEGIN

$act1 : OUT := \varnothing$

$act2 : i := 0$

$act5 : ack := \varnothing$

$act6 : got := \varnothing$

$act8 : b := \varnothing$

END

EVENT send

ANY

j

WHERE

$grd1 : j \in i \mathrel{..} i{+}l$

$grd2 : j \leq n$

$grd3 : j \notin got$

$grd4 : j{-}i \in 0 \mathrel{..} l$

THEN

$act3 : b(j{-}i) := IN(j)$

END

EVENT receive
ANY
j
WHERE
$grd2 : j \in i \,..\, i{+}l$
$grd3 : j{-}i \in dom(b)$
THEN
$act2 : ack := ack \cup \{j\}$
$act3 : OUT(j) := b(j{-}i)$
END

EVENT receiveack
ANY
k
WHERE
$grd1 : k \in ack$
THEN
$act1 : got := got \cup \{k\}$
$act2 : ack := ack \setminus \{k\}$
END

EVENT sliding
ANY
c
WHERE
$grd1 : got \neq \varnothing$
$grd3 : i \in got$
$grd4 : i{+}l < n$

$$grd5 : \left(\begin{array}{l} c \in 0 \,..\, l \nrightarrow D \\ \wedge dom(c) = \{u | u \in 0 \,..\, l{-}1 \wedge u{+}1 \in dom(b)\} \\ \wedge (\forall o \cdot o \in dom(b) \wedge o \neq 0 \Rightarrow o{-}1 \in dom(c) \wedge c(o{-}1) = b(o)) \end{array} \right)$$

THEN
$act1 : i := i{+}1$
$act2 : got := got \setminus \{i\}$
$act3 : ack := ack \setminus \{i\}$
$act5 : b := c$
END

EVENT emptywindow
ANY
c
WHERE
$grd1 : got \neq \varnothing$
$grd2 : i \in got$
$grd3 : i+l \geq n$
$grd4 : i \leq n$
$$grd5 : \left(\begin{array}{l} c \in 0\,..\,l \nrightarrow D \\ \wedge dom(c) = \{u | u \in 0\,..\,l-1 \wedge u+1 \in dom(b)\}) \\ \wedge(\forall o \cdot o \in dom(b) \wedge o \neq 0 \Rightarrow o-1 \in dom(c) \wedge c(o-1) = b(o)) \end{array}\right)$$
THEN
$act1 : i := i+1$
$act2 : got := got \setminus \{i\}$
$act3 : ack := ack \setminus \{i\}$
$act5 : b := c$
END

EVENT completion

WHEN
$grd1 : i = n+1 \wedge got = \varnothing$
THEN
$skip$
END

EVENT loosingchan

ANY
j
WHERE
$grd1 : j \in i\,..\,i+l$
$grd3 : j \notin got$
$grd4 : j-i \in dom(b)$
THEN
$act3 : b := \{j-i\} \lhd b$
END

EVENT loosingack

ANY
k
WHERE
$grd1 : k \in ack$
THEN
$act1 : ack := ack \setminus \{k\}$
END