Utilisation et approches de vérification dans le projet DISPO

J.-P. Bodeveix, M. Filali

IRIT

Nantes

Jeudi 23 septembre 2004

1 Fusion de réseaux de composants

Raisonnement sur la fusion

Utilsation de TLA

- Modélisation des composants et leur fusion.
- Expression de propriétés de la fusion.
- Vérification de propriétés sur l'abstraction.

MODULE *Producteur*

LOCAL INSTANCE *Sequences*

LOCAL INSTANCE *Naturals*

VARIABLES $pc,\ x,\ f$

$Init \triangleq \ \land x = 0 \land f = \langle\rangle \land pc = \text{“p0”}$

$Op1 \triangleq \ \land pc = \text{“p0”} \land x' = x + 1 \land pc' = \text{“p1”} \land f' = f$

$Op2 \triangleq \ \land pc = \text{“p1”} \land x' = x \land pc' = \text{“p0”} \land f' = Append(f,\ x)$

$Next \triangleq \ \lor Op1 \lor Op2$

$Spec \triangleq \ Init \land \Box[Next]_{\langle pc,\ x,\ f\rangle}$

MODULE *OrdoStatique*

LOCAL INSTANCE *Sequences*

VARIABLES f,

$pc1$, x,

a, b, $pc2$, i, o,

sP, sC schedule P, schedule C

$vP \triangleq \langle pc1, x, f \rangle$

$vC \triangleq \langle a, b, pc2, i, o, f \rangle$

$vProcs \triangleq \langle pc1, x, f, a, b, pc2, i, o \rangle$

$privP \triangleq \langle pc1, x \rangle$

$privC \triangleq \langle a, b, pc2, i, o \rangle$

$shared \triangleq \langle f \rangle$

$variables \triangleq \langle i, o, f, x, pc1, a, b, pc2, sP, sC \rangle$

$P \triangleq \text{INSTANCE } Producteur \text{ WITH } pc \leftarrow pc1$

$C \triangleq \text{INSTANCE } Consommateur \text{ WITH } pc \leftarrow pc2$

$Init \triangleq \ \land P!Init \land C!Init \land sP = \text{TRUE} \land sC = \text{FALSE}$

$Next \triangleq$

$\lor \ sP \land \text{ENABLED } \langle P!Next \rangle_{vP} \land P!Next \land \text{UNCHANGED } privC \land sP' = \text{FALSE} \land sC' = \text{TRUE}$

$\lor \ sP \land \neg\text{ENABLED } \langle P!Next \rangle_{vP} \land \text{UNCHANGED } vProcs \land sP' = \text{FALSE} \land sC' = \text{TRUE}$

$\lor \ sC \land \text{ENABLED } \langle C!Next \rangle_{vC} \land C!Next \land \text{UNCHANGED } privP \land sC' = \text{FALSE} \land sP' = \text{TRUE}$

$\lor \ sC \land \neg\text{ENABLED } \langle C!Next \rangle_{vC} \land \text{UNCHANGED } vProcs \land sC' = \text{FALSE} \land sP' = \text{TRUE}$

$Spec \triangleq \ Init \land \Box[Next]_{variables}$

2 Scheduler Abstrait

Principes:

- Suppression des données.
- Abstraction des files internes par des compteurs.
- Suppression des files de sortie.
- Abtraction des files d'entrées par des booléens.

Remarque: Possibilité de réutilisation de l'algorithme de calcul des bornes.

MODULE *Producteur_abs*

LOCAL INSTANCE *Naturals*

VARIABLES pc, abs_f_cpt

$Init \triangleq \ \land abs\_f\_cpt = 0 \land pc = \text{“p0”}$

$Op1 \triangleq \ \land pc = \text{“p0”} \land pc' = \text{“p1”} \land \text{UNCHANGED } abs\_f\_cpt$

$Op2 \triangleq \ \land pc = \text{“p1”} \land pc' = \text{“p0”} \land abs\_f\_cpt' = abs\_f\_cpt + 1$

$Next \triangleq \ \lor Op1 \lor Op2$

MODULE *Producteur_abs*

LOCAL INSTANCE *Naturals*

VARIABLES pc, abs_f_cpt

$Init \triangleq \ \land abs\_f\_cpt = 0 \land pc = \text{“p0”}$

$Op1 \triangleq \ \land pc = \text{“p0”} \land pc' = \text{“p1”} \land \text{UNCHANGED } abs\_f\_cpt$

$Op2 \triangleq \ \land pc = \text{“p1”} \land pc' = \text{“p0”} \land abs\_f\_cpt' = abs\_f\_cpt + 1$

$Next \triangleq \ \lor Op1 \lor Op2$

MODULE *Ordo_abs*

LOCAL INSTANCE *Sequences*
LOCAL INSTANCE *Naturals*
VARIABLES abs_f_cpt, $pc1$, abs_i_empty, $pc2$, gP, gC
$vP \triangleq \langle pc1, abs\_f\_cpt\rangle$
$vC \triangleq \langle abs\_i\_empty, pc2, abs\_f\_cpt\rangle$
$vProcs \triangleq \langle pc1, abs\_f\_cpt, pc2, abs\_i\_empty\rangle$
$privP \triangleq \langle pc1\rangle$
$privC \triangleq \langle pc2, abs\_i\_empty\rangle$
$shared \triangleq \langle abs\_f\_cpt\rangle$
$variables \triangleq \langle abs\_i\_empty, abs\_f\_cpt, pc1, pc2, gP, gC\rangle$

$P \triangleq$ INSTANCE *Producteur_abs* WITH $pc \leftarrow pc1$
$C \triangleq$ INSTANCE *Consommateur_abs* WITH $pc \leftarrow pc2$

$Init \triangleq \land P!Init \land C!Init \land gP = \text{TRUE} \land gC = \text{FALSE}$
$Next \triangleq$
$\lor gP \land (\text{ENABLED } \langle P!Next\rangle_{vP}) \land P!Next \land \text{UNCHANGED } privC \land gP' = \text{FALSE} \land gC' = \text{TRUE}$
$\lor gP \land (\neg\text{ENABLED } \langle P!Next\rangle_{vP}) \land \text{UNCHANGED } vProcs \land gP' = \text{FALSE} \land gC' = \text{TRUE}$
$\lor gC \land (\text{ENABLED } \langle C!Next\rangle_{vC}) \land C!Next \land \text{UNCHANGED } privP \land gC' = \text{FALSE} \land gP' = \text{TRUE}$
$\lor gC \land (\neg\text{ENABLED } \langle C!Next\rangle_{vC}) \land \text{UNCHANGED } vProcs \land gC' = \text{FALSE} \land gP' = \text{TRUE}$
$Spec \triangleq Init \land \Box[Next]_{variables}$

$Inv \triangleq (abs\_f\_cpt <= 1)$

3 Extensions

- Langage de spécification de schedulers,
- Synthèse d'automates à partir de formules logiques.
- Versions temporisées.

4 Composants communicants asynchrone

Etudes

- Réutilisation du calcul de borne pour une version temporisée: une fois que l'on a borné le système, on peut modéliser et vérifier des «deadlines» associés aux requêtes.