



Réalisation Assistée d'Applications Réparties
Examen (14 Novembre 2012)

B.Bérard, E. Borde & F. Kordon

Durée : 2h

Les téléphones portables doivent être éteints et dans les sacs

Chacune des parties de ce sujet doit être traitée sur une copie différente

Avant-propos Toutes vos réponses doivent être claires et justifiées (une réponse non justifiée peut être considérée comme fausse). Les barèmes associés aux questions sont indicatifs. Ne confondez pas « bonne réponse » avec « longue réponse » ;-)

Partie I

Prototypage, génération automatique de programmes (10 points)

1 Sur le cours

Question I.1 (1,5 point) Qu'est ce qu'un « langage pivot » dans un processus de génération de code ? donnez un exemple d'un tel langage (20 lignes).

Question I.2 (1,5 point) Pouvez-vous expliquer la différence d'architecture entre un intergiciel « générique » et un intergiciel « schizophrène » (25 lignes).

2 Génération de programmes

Nous souhaitons générer un programme associé au réseau de Petri de la figure 1. L'invocation d'un outil de calcul des propriétés structurelles identifie les invariants suivants :

$$F_1 \quad p_{10} + p_9 + p_{12} + p_8 + p_1 + p_3 + p_4 + p_5 = cst$$

$$F_2 \quad p_8 + p_7 + p_1 + p_3 + p_4 = cst$$

$$F_3 \quad p_8 + p_5 + p_6 = cst$$

$$F_4 \quad p_5 + p_2 = cst$$

$$F_5 \quad p_{11} + p_{12} + p_{13} = cst$$

Question I.3 (1 point) Peut-on proposer un (ou plusieurs) partitionnement(s) du modèle en automates communicants ? Expliquez brièvement pourquoi et explicitiez ce (ces) partitionnement(s).

Question I.4 (1 point) Indiquez tous les objets de génération associés au(x) partitionnement(s) que vous proposez.

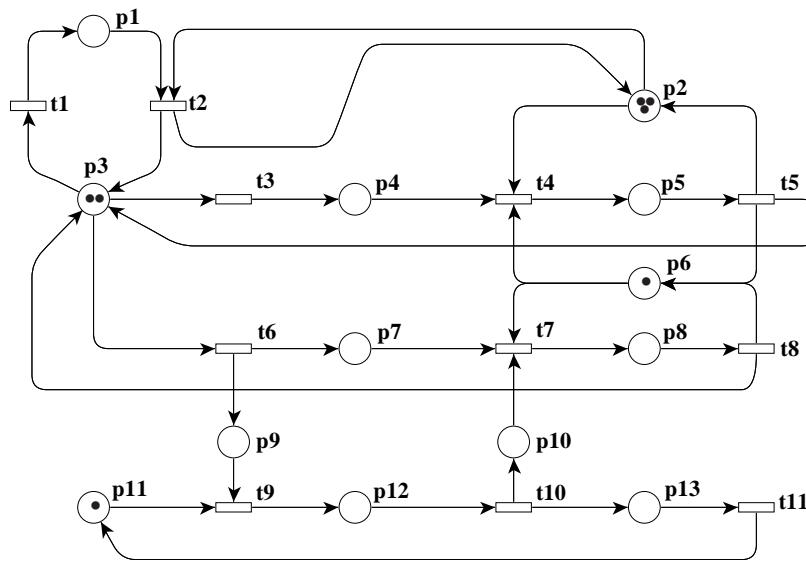


FIGURE 1 – Le réseau de Petri à étudier.

Question I.5 (1,5 point) Nous considérons une approche de génération de code similaire à celle présenté en cours. Indiquez combien de threads comporte l'implémentation de cette spécification (vous devez le faire pour chaque décomposition que vous aurez trouvé) ?

Question I.6 (1 point) On considère une stratégie de gestion « hybride » des ressources basée sur une analyse structurelle de la spécification. Quel partitionnement proposez-vous ? justifiez brièvement votre réponse.

Question I.7 (1 point) On dispose d'une machine à N processeurs. Quelle valeur donnez-vous à N pour exécuter efficacement le code généré ?

Question I.8 (1,5 point) En considérant la valeur N identifiée dans la question I.7 et le partitionnement des ressources de la question I.6, proposez un placement des entités du code généré sur une architecture cible à N processeurs.

Partie II

Sémantique d'exécution des applications réparties (6 points)

Question II.1 - Algèbres de processus.

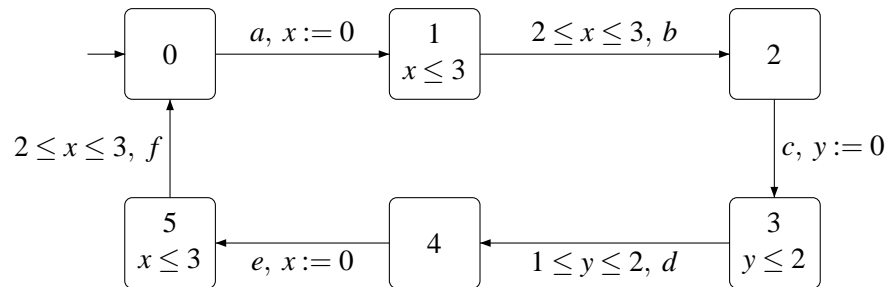
On considère les processus suivants, définis dans l'algèbre de processus du cours par :

$$P \stackrel{\text{def}}{=} b.Q \quad Q \stackrel{\text{def}}{=} a.R \quad R \stackrel{\text{def}}{=} b.Q + c.\mathbf{0} \quad S \stackrel{\text{def}}{=} b.a.T \quad T \stackrel{\text{def}}{=} \epsilon.S + c.\mathbf{0}$$

1. Dessiner les systèmes de transition associés à ces processus.
2. Comparer les processus P et S du point de vue des langages acceptés (en considérant les états P et S comme initiaux et tous les états comme finaux), on donnera ces langages.
3. Comparer P et S du point de vue de la bisimulation (faible).

Question II.2 - Automates temporisés.

On considère l'automate temporisé \mathcal{A} représenté ci-dessous.



1. Pour chaque état i , $0 \leq i \leq 5$, dessiner la zone Z_i obtenue à l'arrivée dans l'état, et la zone maximale Z'_i obtenue en passant du temps dans l'état. Donner les équations de ces zones.
2. Dessiner la zone Z_0^1 obtenue à l'arrivée dans l'état 0 après un tour dans l'automate. Donner son équation.

Partie III

Modélisation en AADL (4 points)

3 Questions de cours

Question III.1 (1 point) Etant donné un thread, quelle est la différence entre un "in out data port" et un "require data access" du point de vue des instants auxquels le code du thread pourra avoir accès à la donnée concernée.

L'utilisation d'un data port est-il bloquant ? Justifier votre réponse.

Question III.2 (1 point) Etant donnée une connection CNX entre deux ports P1 et P2 appartenant chacun à un composant process différent (respectivement Proc1 et Proc2). Supposons que Proc1 et Proc2 sont déployés sur le même processeur. Expliquer comment modéliser le fait que la communication entre P1 et P2 passe sur un bus de communication.

Deux éléments de réponse sont attendus :

1. la spécification de l'architecture matérielle (bus+connection avec le processeur).
2. la spécification du déploiement de la connection logique sur le media de communication.

4 Modélisation AADL

Question III.3 (2 points) Nous allons modéliser en AADL la partie "arrêt d'urgence" d'un robot suiveur de ligne. Nous supposons qu'un thread AADL doit récupérer toutes les 100 millisecondes une valeur de distance fournie par un capteur de distance ("device Sonar") dans le code AADL fourni. Si la valeur de distance est inférieure à une certaine limite, le thread doit émettre un évènement sur un port STOP_ENGINE. Nous ne nous intéressons pas ici à la modélisation de ce comportement, mais simplement à la définition du thread, de ses **propriétés d'ordonnement**, de ses **interfaces** et de la suite de **connections** qui permettent de représenter la capacité du thread à récupérer les valeurs de "distance".

En particulier, nous imposons de réutiliser la spécification matérielle suivante, ainsi que le canevas de spécification système :

Listing 1 – Canevas de Code AADL

```

1  device Sonar
2  features
3    getDistance: provides subprogram access GetSonarSensor;
4    phyPort: requires bus access IC2;
5  end Sonar;
6
7  processor arm
8  features
9    phyPort: requires bus access IC2;
10 end arm;
11
12 bus IC2
13 end IC2;
14
15 subprogram GetSonarSensor
16 features
17   portNb: in parameter Int;
18   distance : out parameter Int;
19 end GetSonarSensor;
20
21 system implementation robot.Impl
22 subcomponents
23   nxt_cpu      : processor arm;
24   PROC        : process Proc.Impl;
25   DistanceSensor: device Sonar;
26   b: bus IC2;
27 connections
28   cnx1: bus access b -> nxt_cpu.phyPort;
29   cnx2: bus access b -> DistanceSensor.phyPort;
30   — TO BE COMPLETED WITH CONNECTIONS
31   — BETWEEN THE INTERFACES OF THE DEVICE
32   — SUBCOMPONENT AND THE PROCESS SUBCOMPONENT
33 end robot.Impl;
34
35 process Proc
36 features
37   — TO BE COMPLETED WITH THE DEFINITION
38   — OF INTERFACES TO RELAY THE CONNECTION
39   — FROM THREAD TO DEVICE
40 end Proc

```

Listing 2 – Suite du Canevas de Code AADL

```

1  process implementation Proc.impl
2  subcomponents
3    distThread: thread distanceMonitor;
4  connections
5    — TO BE COMPLETED WITH THE DEFINITION
6    — OF CONNECTIONS BETWEEN THE INTERFACES OF
7    — THE THREAD AND INTERFACES OF THE
8    — PROCESS PARENT COMPONENT.
9  end Proc.impl;
10
11 thread distanceMonitor
12 features
13   — TO BE COMPLETED WITH THE DEFINITION
14   — OF THE INTERFACES OF THE THREAD
15 properties
16   — TO BE COMPLETED WITH THE DEFINITION
17   — OF THE SCHEDULING PROPERTIES OF THE
18   — THREAD
19 end distanceMonitor

```