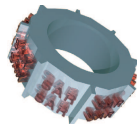


Master UPMC Sciences et Technologies, mention Informatique  
Spécialité Systèmes et Applications Réparties

## Réalisation Assistée d'Applications Réparties



### Examen

B.Bérard, F. Kordon & G. Lasnier

16 Novembre 2011

*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 point)** Existe-t-il une «technique de prototypage» ne requérant pas ou peu d'instrumentation ? si oui, laquelle ? Justifiez brièvement votre réponse.

**Question I.2 (1 point)** Expliquez la notion de «double neutralité» lorsqu'on évoque les intergiciels «schizophrènes».

##### 2 Génération de programmes

**Question I.3 (1 point)** Nous souhaitons générer un programme associé au réseau de Petri de la figure 1. L'invocation d'un outil de calcul des propriétés structurelles nous ramène le calcul des invariants suivants :

$$F_1 \quad h + g + o + k = cst$$

$$F_2 \quad a + o + e + k = cst$$

$$F_3 \quad a + b + d + j + h + g + f = cst$$

$$F_4 \quad 2 \times a + b + d + j + e + f = cst$$

$$F_5 \quad j + i + h + g = cst$$

$$F_6 \quad a + j + i + e = cst$$

$$F_7 \quad t + s + r + q = cst$$

$$F_8 \quad o + n + m + l = cst$$

$$F_9 \quad o + n + p = cst$$

$$F_{10} \quad a + b + c + d = cst$$

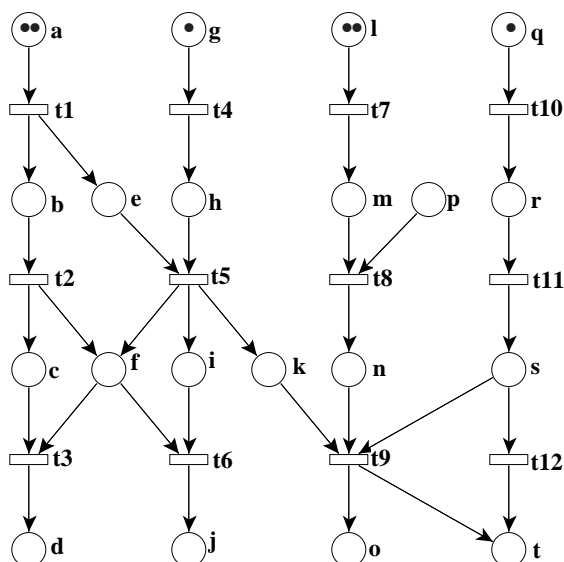


FIGURE 1 – Le réseau de Petri à étudier.

**Question I.3 (1 point)** Une analyse préliminaire de ces invariants nous permet-elle d'en supprimer certains ? Justifiez votre réponse

**Question I.4 (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.5 (0,5 point)** Comparez ce(s) partitionnement(s). Quels critères peuvent permettre d'en choisir un plutôt qu'un autre (s'il y en a plusieurs) ?

**Question I.6 (1 point)** Indiquez tous les objets de génération associés au(x) partitionnement(s) que vous proposez.

**Question I.7 (1 point)** Pouvez-vous indiquer combien de threads aura l'implémentation de cette spécification (vous devez le faire pour chaque décomposition que vous aurez trouvée) ?

**Question I.8 (1 point)** Si on considère une stratégie de gestion «hybride» des ressources, quel partitionnement proposez-vous ? justifiez brièvement votre réponse.

**Question I.9 (1 point)** Quel placement des entités du code généré suggérez-vous sur une machine tri-processeurs ?

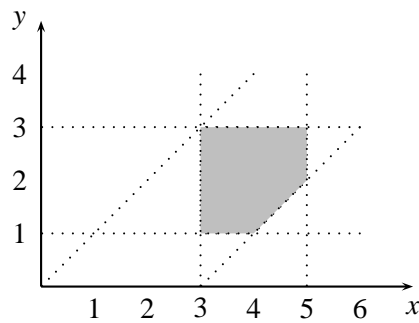
**Question I.10 (0,5 point)** Pouvez-vous dire si la détection de fin d'exécution est possible sur le programme généré ? dans quelles conditions peut-elle survenir ?

## Partie II

### Sémantique d'exécution des applications réparties (6 points)

#### 3 Automates temporisés

**Question II.1 - Automates temporisés.** Pour l'ensemble d'horloges  $X = \{x, y\}$ , on considère la zone  $Z$  dessinée ci-dessous.



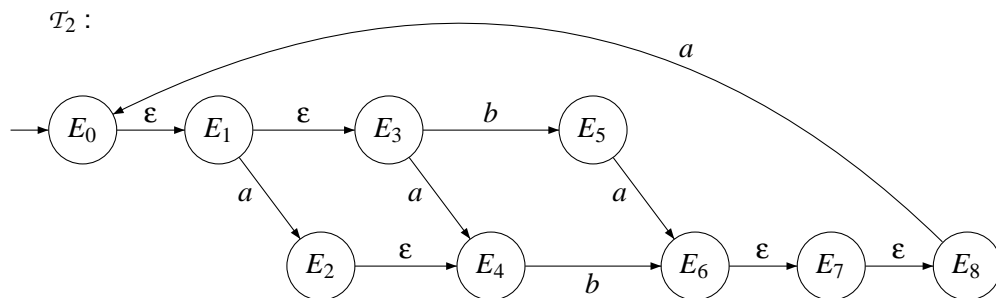
1. Donner une conjonction de contraintes qui la définit. Donner les contraintes définissant le futur de  $Z$ .
2. Construire un automate temporisé  $\mathcal{A}$ , sans garde diagonale, avec la propriété suivante :  $\mathcal{A}$  comporte un état  $q$  pour lequel l'ensemble des valuations  $v$  obtenues en arrivant dans  $q$  est exactement  $Z$ .

#### Question II.2 - Algèbres de processus.

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

$$F_1 \stackrel{\text{def}}{=} a.F_2 + b.F_3 \quad F_2 \stackrel{\text{def}}{=} b.F_4 \quad F_3 \stackrel{\text{def}}{=} a.F_4 \quad F_4 \stackrel{\text{def}}{=} a.F_1$$

1. Dessiner le système de transition  $\mathcal{T}_1$  associé.
2. Comparer les systèmes  $\mathcal{T}_1$  et  $\mathcal{T}_2$ , où  $\mathcal{T}_2$  est le système représenté ci-dessous.



## Partie III

### Modélisation en AADL (4 points)

#### 4 Questions de cours

**Question III.1 (0,5 point)** Quelles sont les motivations du langage AADL ?

**Question III.2 (0,5 point)** Donner deux types de mécanismes qui permettent d'enrichir une description architecturale en AADL ?

On considère le système suivant : L'application contient deux noeuds s'exécutant sur la même plate-forme. Le premier noeud envoie un message au second toutes les 300 ms. A la réception du message, le second noeud (port : 12002) l'analyse et déclenche un traitement.

Le paquetage Behaviors contient les routines des sous-programmes Ada95 implantant les comportements des différents threads du système.

Ainsi, les routines Do\_Something\_Spg et Do\_Something\_Else\_Spg implantent le comportement des threads P et Q.

Nous donnons le modèle AADL incomplet (listing 1) du système décrit :

---

```
-- data
data Simple_Type
properties
  Data_Model::Data_Representation => Integer;
end Simple_Type;

-- subprogram
subprogram Do_Something_Spg
features
  Data_Source : out parameter Simple_Type;
end Do_Something_Spg;

subprogram implementation Do_Something_Spg.Impl
properties
  -- TBC
  -- TBC
end Do_Something_Spg.Impl;

-- threads
thread P
features
  -- TBC
end P;

thread implementation P.Impl
calls {
  P_Spg : subprogram Do_Something_Spg.Impl;
};
connections
  parameter P_Spg.Data_Source -> Data_Source;
properties
  -- TBC
  -- TBC
  -- TBC
end P.Impl;

thread Q
features
```

```

    -- TBC
end Q;

thread implementation Q.Impl
calls {
    Q_Spg : subprogram Do_Something_Else_Spg;
};
connections
    parameter Data_Sink -> Q_Spg.Data_Sink;
properties
    -- TBC
    Period                => 10 Ms;
    Deadline               => 10 Ms;
end Q.Impl;

-- process
process A
features
    Out_Port : out event data port Simple_Type;
end A;

process implementation A.Impl
subcomponents
    Sender      : thread P.Impl;
connections
    event data port Sender.Data_Source -> Out_Port;
end A.Impl;

process B
features
    In_Port : in event data port Simple_Type;
end B;

process implementation B.Impl
subcomponents
    Receiver      : thread Q.Impl;
connections
    event data port In_Port -> Receiver.Data_Sink;
end B.Impl;

-- processor
processor x86_Proc
features
    ETH : requires bus access Ethernet_Bus;
properties
    Deployment::Location                => "127.0.0.1";
    Deployment::Execution_Platform => Native;
end the_processor;

-- bus
bus Ethernet_Bus
properties
    Deployment::Transport_API => BSD_Sockets;
end Ethernet_Bus;

-- system
system UNKNOW
end UNKNOW;

system implementation UNKNOW.Native

```

```

subcomponents
  Node_A : process A.Impl;
  Node_B : process B.Impl { -- TBC                               };

  The_CPU : processor x86_Proc;
  The_BUS : bus Ethernet_Bus;
connections
  bus access The_Bus -> The_CPU.ETH;

  -- TBC

properties
  -- TBC
  -- TBC
end UNKNOW.Native;

```

---

Listing 1 – Modèle AADL

**Question III.3 (0,5 points)** A l'aide de vos connaissances et en analysant le modèle AADL donné (notamment les connexions spécifiées), compléter l'interface et l'implantation des threads P et Q. On rappelle qu'un évènement se modélise à l'aide des event port dans le langage AADL.

**Question III.4 (0,5 points)** A l'aide des informations et du modèle AADL à votre disposition, compléter l'implantation du sous-programme Do\_Something\_Spg et donner l'interface du sous-programme Do\_Something\_Else\_Spg.

**Question III.5 (0,25 points)** A l'aide de la propriété Port\_Number de l'ensemble de propriétés Deployment, compléter les informations de déploiement relatives au noeud Node\_B dans le système UNKNOW.Native.

**Question III.6 (0,5 points)** Compléter la section connections du système UNKNOW.Native. Nous rappelons que pour que le modèle soit "légal" au sens AADL, il est nécessaire d'associer un bus à une connexion AADL.

**Question III.7 (0,5 points)** Finalement, donner et compléter la propriété manquante pour compléter la description des composants matériels et logiciels qui constituent le système UNKNOW.Native. A quoi sert cette propriété ?

**Question III.8 (0,25 points)** Après analyse du modèle AADL complété, donner un exemple de cas d'utilisation pour le système UNKNOW.Native.