ENSEIRB
Année 2000-2001

3^{eme} année
Option Génie Logiciel

Lundi 12 février, 10 heures - 12 heures
Tous documents autorisés.

Ingénierie des Systèmes Critiques
Alain Griffault

SUJET + CORRIGE

---

1 Le cahier des charges

1.1 Le cadre général

L'objectif de l'étude est la réalisation d'un contrôleur pour la gestion des visiteurs d'un musée. Le musée est composé de salles et de portes. Des portes permettent le passage (dans les deux sens) soit d'une salle à une autre, soit d'une salle vers l'extérieur. Chaque salle à une capacité maximale de visiteurs simultanés, et le contrôleur est chargé de veiller que ce nombre (ou un autre plus petit) ne soit jamais dépassé.

1.2 L'exemple étudié

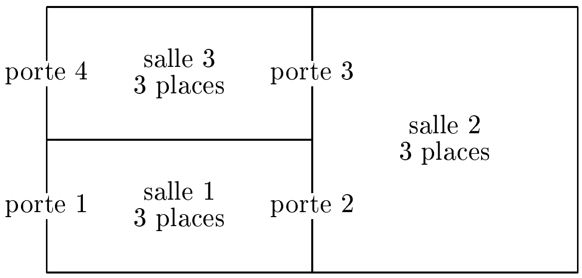

FIG. 1 – Le musée

2 Modélisation du musée en Mec

Dans un premier temps, le système est étudié sans contrôleur.

Question 2.1 (1 point) *Dessinez le système de transition pour les trois salles du musée. Vous utiliserez* `in` *et* `out` *comme étiquettes pour les transitions.*

Réponse à la question 2.1: Une abstraction possible :

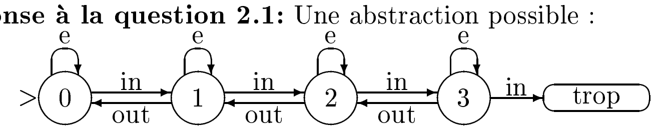

Figure 2: une salle

Question 2.2 (1 point) *Dessinez le système de transition d'une porte. Vous utiliserez* `ouvrir`, `fermer`, `ouverte` *et* `fermée` *comme étiquettes pour les transitions.*

Réponse à la question 2.2: Une abstraction possible :

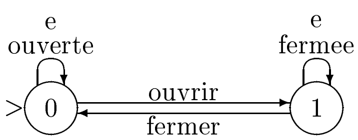

Figure 3: une porte

Question 2.3 (2 points) *Donnez le système de synchronisation du système global.*

Réponse à la question 2.3:

porte 1	salle 1	porte 2	salle 2	porte 3	salle 3	porte 4
ouverte	in	e	e	e	e	e
ouverte	out	e	e	e	e	e
e	out	ouverte	in	e	e	e
e	in	ouverte	out	e	e	e
e	e	e	out	ouverte	in	e
e	e	e	in	ouverte	out	e
e	e	e	e	e	out	ouverte
e	e	e	e	e	in	ouverte
ouvrir	e	e	e	e	e	e
fermer	e	e	e	e	e	e
e	e	ouvrir	e	e	e	e
e	e	fermer	e	e	e	e
e	e	e	e	ouvrir	e	e
e	e	e	e	fermer	e	e
e	e	e	e	e	e	ouvrir
e	e	e	e	e	e	fermer

Le conservateur du musée décide de salarier un contrôleur. Celui-ci peut d'une part ouvrir et fermer les portes du musée ; d'autre part observer les (ou certains) déplacements des visiteurs.

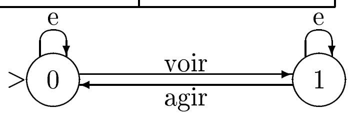

Figure 4: un contrôleur

Le conservateur place le contrôleur comme indiqué sur le schéma, afin qu'il voit les portes 1, 2 et 4, mais pas la porte 3. Le conservateur lui fournit un boîtier de commandes muni de quatre boutons pour commander les quatre portes. On suppose que le boîtier ne permet pas d'appuyer simultanément sur deux boutons.

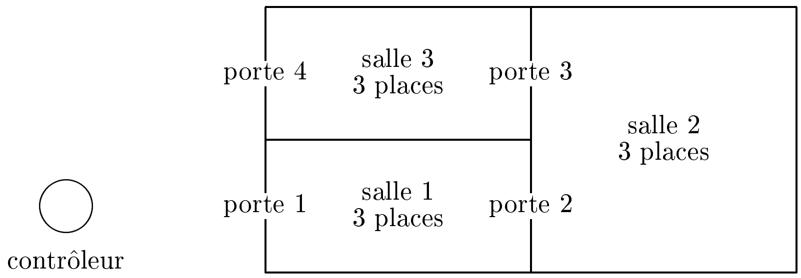

Fig. 5 – Le musée et le contrôleur

Question 2.4 (2 points) *Donnez le nouveau système de synchronisation du système global.*

Réponse à la question 2.4:

contrôleur	porte 1	salle 1	porte 2	salle 2	porte 3	salle 3	porte 4
voir	ouverte	in	e	e	e	e	e
voir	ouverte	out	e	e	e	e	e
voir	e	out	ouverte	in	e	e	e
voir	e	in	ouverte	out	e	e	e
e	e	e	e	out	ouverte	in	e
e	e	e	e	in	ouverte	out	e
voir	e	e	e	e	e	out	ouverte
voir	e	e	e	e	e	in	ouverte
agir	ouvrir	e	e	e	e	e	e
agir	fermer	e	e	e	e	e	e
agir	e	e	ouvrir	e	e	e	e
agir	e	e	fermer	e	e	e	e
agir	e	e	e	e	ouvrir	e	e
agir	e	e	e	e	fermer	e	e
agir	e	e	e	e	e	e	ouvrir
agir	e	e	e	e	e	e	fermer

3 Synthèse du contrôleur

Le conservateur constate que malgré la présence de son employé, les normes de sécurité ne sont pas toujours respectées (il y a quelquefois trop de visiteurs dans les salles). Le conservateur du musée souhaite que le contrôleur fasse en sorte que dans chacune des pièces du musée, le nombre de visiteurs soit limité à un. Le contrôleur dit alors à son patron que pour cela, il a besoin d'un mec supplémentaire.

3.1 Les préliminaires

Le système global est assimilé comme un jeu à deux joueurs. D'une part le contrôleur qui doit ouvrir et fermer les portes à bon escient ; d'autre part les visiteurs qui vont essayer de faire licencier le contrôleur.

Question 3.1 (2 points) *Donnez les expressions qui permettent de calculer les ensembles suivants :*

Réponse à la question 3.1:

```
\* gagneX : les etats ou X doit jouer et a gagne (la partie est finie) *\
\* perduX : les etats ou X doit jouer et a perdu (la partie est finie) *\
\* coupX : les transitions que peut jouer X *\
dead := * - src(*);
coupControleur := !label[1]='agir';
coupVisiteurs  := * - coupControleur;
// GagneControleur := n'est pas vraiment définissable;
GagneVisiteurs := ((!state[3]#'0'/\!state[3]#'1')\/
                   (!state[5]#'0'/\!state[5]#'1'))\/
                  (!state[7]#'0'/\!state[7]#'1'));
PerduControleur := GagneVisiteurs;
// PerduVisiteurs := n'est pas vraiment définissable;
```

3.2 Le calcul des décisions du contrôleur

Question 3.2 (2 points) *Donnez la fonction au point fixe qui permet de calculer les fermetures et les ouvertures des portes, et l'appel utilisé concrètement.*

Réponse à la question 3.2: La fonction est similaire à celle de la recherche des coups gagnant dans un jeu à deux joueurs. Une petite différence : dans un jeu, on cherche à gagner le plus rapidement possible, ce qui se traduit par un plus petit point fixe, pour un contrôleur, l'objectif est de ne pas perdre, il s'agit donc d'un plus grand point fixe.

```
function ActionsControleurPGF(perduX:state;
                              gagneY:state;
                              coupX:trans;
                              coupY:trans)
    return Xgagnante:_trans;
var Yperdante:_trans
begin
Xgagnante = ((rtgt(src(Yperdante))/\coupX)-(rtgt(src(coupY-Yperdante))/\coupX))
            - (rtgt(gagneY)/\coupX);
Yperdante = (rtgt(src(Xgagnante)) /\ coupY)
            - (rtgt(perduX)/\coupY);
end.
```

Cette fonction est utilisée de la façon suivante :

```
action := ActionsControleurPGF(PerduControleur,GagneVisiteurs,coupControleur,coupVisiteurs);
```

Question 3.3 (2 points) *Donnez la fonction au point fixe qui permet de calculer l'ensemble des transitions (contrôleur et visiteurs)* contrôlées*, et l'appel utilisé concrètement.*

Réponse à la question 3.3: Le système d'équation est identique, mais le résultat est l'union des deux ensembles.

```
function ControleurPGF(perduX:state;
                       gagneY:state;
                       coupX:trans;
                       coupY:trans)
    return res:_trans;
```

```
var Xgagnante:_trans;
    Yperdante:_trans
begin
Xgagnante = ((rtgt(src(Yperdante))/\coupX)-(rtgt(src(coupY-Yperdante))/\coupX))
            - (rtgt(gagneY)/\coupX);
Yperdante = (rtgt(src(Xgagnante)) /\ coupY)
            - (rtgt(perduX)/\coupY);
res = Xgagnante \/ Yperdante
end.
```

Cette fonction est utilisée de la façon suivante :

```
action := ControleurPGF(PerduControleur,GagneVisiteurs,coupControleur,coupVisiteurs);
```

Question 3.4 (2 points) *Expliquez pourquoi ce système (avec ces hypothèses) n'est pas contrôlable si initialement toutes les portes sont ouvertes.*

Réponse à la question 3.4: Soit le scénario suivant :

1. Le contrôleur voit une personne entrer dans la salle 1,
2. le contrôleur est obligé de fermer la porte 1, sinon il ne pourra pas empêcher une seconde personne dans la salle 1,
3. le contrôleur voit une personne entrer dans la salle 2,
4. le contrôleur est obligé de fermer la porte 2, sinon il ne pourra pas empêcher une seconde personne dans la salle 2,
5. la seconde personne entre dans la salle 3 (le contrôleur ne le voit pas),
6. le contrôleur voit la première personne entrer dans la salle 3,
7. **il y a plus d'une personne dans la salle 3**

Ce scénario ne peut pas être interdit par le contrôleur, le système n'est donc pas contrôlable.

Question 3.5 (2 points) *Proposez une revendication (les portes restant initialement ouvertes) que pourrait faire le contrôleur auprès du conservateur, afin qu'il puisse faire son travail. Justifiez.*

Réponse à la question 3.5: Afin d'éviter le scénario précédent, et si l'on veut que les portes soient initialement ouvertes, deux revendications sont possibles :

- que le conservateur du musée fournisse un boîtier à partir duquel il est possible d'agir sur plusieurs portes simultanément.
- que le conservateur autorise au moins deux personnes dans chaque salle.

Le contrôleur peut également faire d'autres demandes qui lui permettront d'être plus efficace, afin que plus de personnes puissent visiter le musée.

- que le conservateur mette un miroir dans la porte 4 afin que le contrôleur puisse voir également les mouvements à travers la porte 3.
- que le conservateur remplace chacune des portes par un couple de portes orientées (deux portes tourniquet par exemple), et que le boîtier lui permette d'agir simultanément sur un sous-ensemble quelconque des huit portes.

4 Modélisation du musée en AltaRica

Question 4.1 (4 points) *Donnez un modèle AltaRica de votre dernier système.*

Réponse à la question 4.1: Il y a évidement de nombreuses réponses à cette question, en voici une :

```
const Maximum = 4 ;

domain Capacite = [0,Maximum] ;

node Salle
  flow
    nb : Capacite;
  state
    present : Capacite;
  event
    in, out;
  trans
    present < Maximum |- in -> present := present + 1;
    present > 0 |- out -> present := present - 1;
  assert
    present = nb;
  extern initial_state = present = 0;
edon

node main
  sub
    S1, S2, S3 : Salle;
  flow
    p1, p2, p3, p4 : bool;
  event
    in1, out1, from1to2, from2to1, from2to3, from3to2, in3, out3;
  trans
    p1 |- in1, out1 -> ;
    p2 |- from1to2, from2to1 -> ;
    p3 |- from2to3, from3to2 -> ;
    p4 |- in3, out3 -> ;
  sync
    <in1,S1.in>;
    <out1,S1.out>;
    <from1to2,S1.out,S2.in>;
    <from2to1,S1.in,S2.out>;
    <from2to3,S2.out,S3.in>;
    <from3to2,S2.in,S3.out>;
    <in3,S3.in>;
    <out3,S3.out>;
  assert
    p1 = (S1.nb < 2);
    p2 = (((S1.nb < 2) & (S2.nb < 2)) | (S1.nb = 0) | (S2.nb < 2));
    p3 = (((S2.nb < 2) & (S2.nb < 2)) | (S2.nb = 0) | (S2.nb < 2));
    p4 = (S2.nb < 2);
edon
```