Final LO22/AI20 – P2022

Partie I (Sûreté de Fonctionnement) :

L'ERTMS (Système Européen de Management des trains et du trafic) est la norme européenne qui s'impose pour le système de contrôle-commande et signalisation des lignes nouvelles et lors de remplacement des systèmes existants. L'ERTMS se compose d'un système de contrôle-commande (ETCS) et d'un système de communication radiofréquence dédié, adapté du standard GSM (GSM-R) reposant sur le concept de cantons fixes (zones de sécurité fixes dans lesquelles il ne peut y avoir qu'un train). Ces spécifications ont engendré les déclinaisons suivantes :

ETCS (de niveau) 1 : composé uniquement de système de contrôle-commande ETCS qui vise à gérer le trafic ferroviaire. Des consignes sont transmises ponctuellement au train uniquement par les eurobalises.

ETCS (de niveau) 2 : composé du système de contrôle-commande ETCS et de GSM-R qui sert à communiquer entre les trains et les centrales d'exploitation du réseau ferré. Les consignes de signalisation ne sont plus transmises par les eurobalises (cf. Figure 1) mais de manière permanente via le réseau GSM-R. Via ce réseau, le train communique constamment sa position (qu'il détermine avec un odomètre : dispositif calculant en embarqué, à l'aide d'informations de rotation des roues, la position du train, recalée périodiquement grâce aux eurobalises) au centre de contrôle au sol (appelé RBC : Radio Block Centre) qui lui communique en retour les actions à effectuer (vitesse, arrêt, ...).

ETCS niveau 3 : il est encore en cours de développement. La communication est encore basée sur le GSM-R. Les équipements sol étant réduits à l'extrême, cela nécessite le développement d'un moyen embarqué de contrôle de l'intégrité des convois.

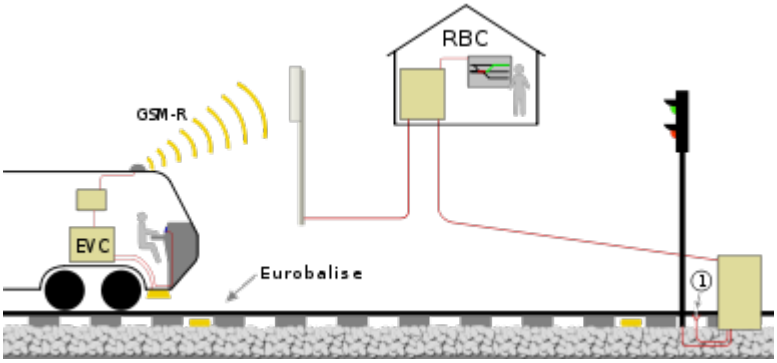

Figure 1. ETCS niveau 2

Dans cet exercice, on s'intéresse à la vérification du respect des exigences de sûreté de fonctionnement par les systèmes ETCS niveau 2.

On s'intéresse plus particulièrement au RBC (Radio Block Centre) qui représente l'équipement fixe de signalisation informatisée qui gère l'espacement entre 2 circulations. Notre objectif est de concevoir un RBC qui vérifie les exigences de sécurité définies par les normes. Dans cette étude, les normes exigent une indisponibilité asymptotique U_{RBC} <0.0275 pour le RBC. Un RBC est constitués essentiellement de :

- Une carte CPU
- Un bus

- Un FPGA
- Un système d'alimentation

Tous ces composants sont nécessaires au fonctionnement du RBC, toute défaillance de l'un d'entre eux entraîne la défaillance du RBC.

Les données de fiabilité et de maintenablité pour chaque composant sont les suivantes :

Composant	MTTF (h)	MTTR (min)
CPU	1.35 * 10^2	300
Bus	2.25 * 10^2	150
FPGA	3.33 * 10^2	150
Système d'alimentation	5.5 * 10^2	300

Le MTTR est le temps moyen jusqu'à réparation (temps de réparation + temps d'attente entre la défaillance et le début de la réparation) et le MTTF est le temps moyen jusqu'à la défaillance. On suppose aussi que les MTTF et les MTTR sont constants au cours du temps et que les lois de défaillances et de réparation des composants sont exponentielles (λ et μ constants).

1) Exprimer le MTTR et le MTTF d'un composant unique en fonction des taux de défaillance et de réparation λ et μ de ce composant.
2) Calculer les disponibilités asymptotiques (à l'infini) de chaque composant avec une précision de 4 chiffres après la virgule.
3) Donner l'arbre de défaillance dont l'événement sommet est la défaillance du RBC et son diagramme de fiabilité équivalent du RBC.
4) Calculer la disponibilité asymptotique totale A_{RBC} du RBC à partir des disponibilités asymptotiques de ses composants (1 CPU + 1 Bus + 1 FPGA + 1 Système d'alimentation). On suppose que les défaillances des composants sont indépendantes. Satisfait elle l'exigence sur l'indisponibilité du RBC : U_{RBC} <0.0276 des normes de sécurité ?
5) Pour satisfaire aux exigences de sécurité, on propose une architecture C1 du RBC avec un vote 2/4 pour les CPUs et une deuxième architecture C2 du RBC avec un vote 3/5 pour les CPUs. Calculer les disponibilités asymptotiques des architectures C1 et C2. Quelle architecture du RBC permet de satisfaire aux exigences de sécurité U_{RBC} <0.0276 ?
6) On s'intéresse maintenant au système complet (1 RBC + 1 GSM-R + 1 Conducteur, tous trois nécessaires au fonctionnement du système). En supposant que la probabilité d'erreur du conducteur est constante et égale à 0.0001 et que le MTTF du GSM-est 2*10^2 (h) et qu'il a un MTTR=300 mn. Calculer l'indisponibilité asymptotique du système complet $U_{Système}$ en considérant l'architecture C1 pour les CPUs du RBC. Satisfait elle l'exigence $U_{Système}$ <0.05 des normes de sécurité ?
7) Calculer le facteur d'importance de Birnbaum pour chaque élément et proposer une solution pour satisfaire l'exigence des normes de sécurité.

Partie II. Méthodes formelles

Exercice 1 (Composition de relations) :

On considère les deux relations R1 et R2 : N↔N (N est l'ensemble des entiers naturels) tel que :

R1 = {1↦4, 2↦3, 3↦2,3↦3, 4↦1}
R2 = {1↦2, 2↦3, 3↦4}

1) Peut-on dire que R1 et R2 sont des fonctions ?
2) Calculer :
   i) R1;R2 et R2;R1. Que peut-on conclure ?
   ii) R1;R1 et R1;R1;R1. Que peut-on conclure ?

Exercice 2 (Obligations de preuves):

On considère la machine "RESERVATION » décrite ci-dessous :

```
MACHINE
       Reservation

VARIABLES
       n_rsrc

INVARIANT
       n_rsrc : 0..100

INITIALISATION
       n_rsrc := 100

OPERATIONS
reserver =
    PRE  n_rsrc > 0
    THEN
         n_rsrc := n_rsrc - 1
    END
;
liberer =
    PRE  n_rsrc < 100
    THEN
         n_rsrc := n_rsrc + 1
    END
;
bb <-- disponibilite =
    bb := bool(0 < n_rsrc)
END
```

1) Écrire et simplifier les obligations de preuves liées à :
   i) L'initialisation.
   ii) L'opération reserver.
   iii) L'opération liberer.

Exercice 3 (Implémentation d'ensembles):

On considère la machine suivante :

```
MACHINE
        M1
SETS
        ETAT
;       PROCESSUS

VARIABLES
        ...
END
```

Proposer une implémentation pour cette machine. L'objectif de cet exercice étant uniquement de proposer une implémentation des ensembles définis dans la clause SETS sans se soucier des autres clauses de la machine.

Exercice 4 :

Écrire et simplifier les substitutions suivantes :

i) [x :=x-1 II x :=x+1] (1<= x <= 10)
ii) [x >0 I x :=x-1] (-1 <= x <= 1)
iii) [@z. (z>0 => x :=x+z)] (3 <= x)
iv) [ANY vv WHERE vv>xx & vv:NAT THEN xx:=vv END] (xx:ss)

Exercice 5 :

On considère la machine suivante :

```
MACHINE
                LO22
SETS
        PERSONNES
CONCRETE_CONSTANTS
        pp,dernier,ff
PROPERTIES
        pp : PERSONNES /\ dernier : NAT1 /\ ff : 0..3 -> NAT1
VALUES
        .....
```

Compléter la clause « VALUES » de cette machine.

Partie III (V&V) :

Exercice 1 (Tests de couvertures) :

On considère le code suivant :

```
lire(b,c,x)
if b<c
then begin
      d :=2*b
      f :=3*c
      if x>=0
      then begin
            y := x
            e := c
            if (y=0)
            then begin
                  a :=f-e
                  while d<a
                        begin
                        d:=d+2
                        end
                  end
            end
      else begin
            b:=b-1
            end
      end
```

1. Donnez un graphe de contrôle associé à ce code source et calculer sa complexité cyclomatique.
2. Votre graphe de contrôle a-t-il des possibilités de réduction ? Si oui, réduisez votre graphe de contrôle.

Exercice 2 :

Soit le programme suivant :

```
Var     x,y,j,i     : Integer ;
        B           : boolean ;
Begin
        j := 0 ;
        repeat
        j := j + 1 ; y := 0 ; B := False ;
                    writeln('donnez un entier x') ;
                    readln(x) ;
                    If x < 0 then
                              x = abs(x) + 1 ;
                    elseif x = 0 then
                              B := True
                    end ;
                    for i = 1 to x
                              y := y + i
                    end ;
        Until B ;
        writeln('résultat =',y) ;
End
```

1. Construire le graphe de contrôle associé à ce programme.
2. Introduire le flot de donnée et étudier la validité du programme.
3. Calculer le nombre cyclomatique de cette procédure.
4. Donner les jeux minimaux de valeurs permettant de couvrir :

- Les instructions
- Les conditions
- Les branches
- Les 1-chemins
- Les 2-chemins

5. Combien de jeu minimal faut-il pour couvrir les 3-chemins.

(**Indication :** pensez à trouver tous les chemins théoriques et procéder par élimination des chemins impossibles.)