Grenoble INP –Ensimag– 3A
Année 2008-2009

Test et Tolérance aux Fautes

Examen du 4 février 2008 – Durée 1h30.
Tous documents autorisés. Les deux parties doivent être rédigées sur des feuilles séparées (Tolérance aux fautes d'un côté, Test de l'autre). Le barème est indicatif.

Partie 1 - Tolérance aux fautes (6 points)

Question 1. *1. Expliquer les raisons et les conditions qui induisent le blocage dans le protocole en 2 phases.*

Question 2. *Lister les types de fautes qui peuvent être tolérées/pas tolérées par le protocole en 2 et en 3 phases.*

Question 3. *Décrivez la configuration et le protocole de généraux byzantins capable de tolérer 2 traîtres, dans le cas spécifique ou il n'y a pas de coordinateur.*

Partie 2 - Test (14 points)

Couverture structurelle de programme

On considère le programme suivant, qui calcule la racine carrée d'un nombre p fourni en entrée ($0 < p < 1$), avec une approximation e ($0 < e < 1$) également fournie en entrée. x est l'approximation courante, d est un écart dichotomique.

```
1. begin
2.    float p, e, x=0, d=1, c, t ;
3.    input (p,e)
4.    c=2*p ;
5.    if c>=2) {
6.          output («  Saisie incorrecte ») ; }
7.    else {
8.          while (d>e) {
9.                d=d/2 ;t=c-(2*x+d) ;
10.               if (t>=0) {
11.                     x=x+d ;
12.                     c=2*(c-(2*x+d))) ;}
13.               else {
14.                     c=2*c ;}
15.         }
16.         output (« Rac (p) = »,x) ;
17.   }
18.end
```

Question 4. *Donnez le flot de données associé à ce programme.*

Question 5. *Donnez un jeu de test simple T_1 qui couvre toutes les branches de ce programme.*

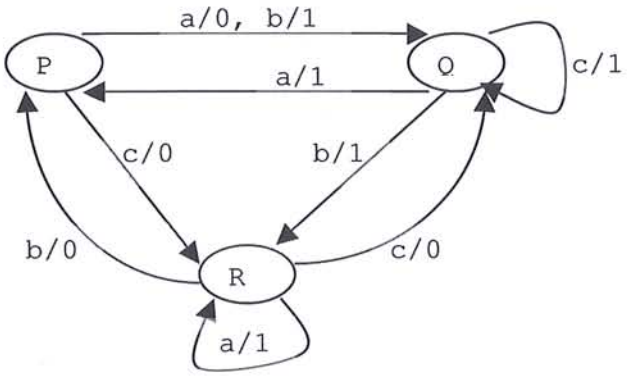
Question 6. *Étendez ce jeu de test pour satisfaire le critère de couverture « all-uses ». Soit T_2 ce jeu de test.*

Question 7. *Le programme ci-dessus contient une faute : les lignes 11 et 12 devraient être inversées. Est-ce que T_1 et T_2 conduisent à une défaillance ?*

Question 8. *Plus largement, est-ce que tout test satisfaisant le critère « all-uses » révèle cette défaillance ?*

Test à base d'automates

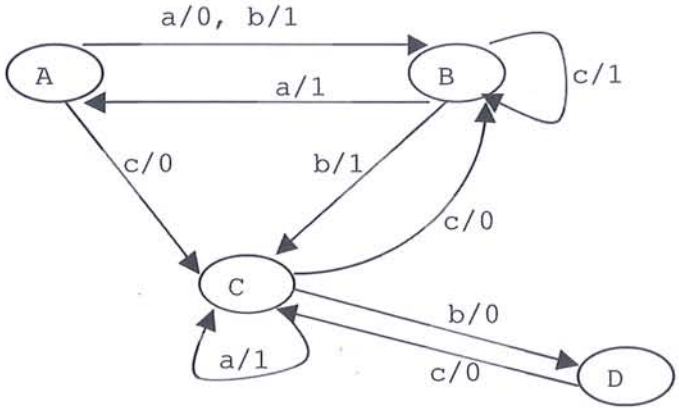
On considère l'automate suivant comme la spécification d'un système à tester.



On suppose que l'implantation est équivalente à un automate n'ayant pas plus de 3 états, et dispose d'une réinitialisation fiable notée r.

Question 9. *Ecrivez une séquence de test T aussi courte que possible pour cet automate sous la forme de la concaténation de deux ensembles de séquences P et Z qu'on explicitera. Cette séquence devra être capable de détecter toute faute d'implantation.*

On considère maintenant l'implantation suivante.



Question 10. *Votre séquence de test T trouve-t-elle l'erreur d'implantation ? Pourquoi ?*

Question 11. *Comment faudrait-il étendre T en une séquence T' trouvant toutes les fautes d'implantation possibles avec des machines ayant au plus 4 états ? Ecrivez T' sous une forme analogue à celle utilisée pour T.*