

UPMC – Master informatique – M2-STL – UE Tests
Examen – Mars 2011

1 MC/DC

Soient c_1 , c_2 et c_3 3 variables booléennes. Soient les deux décisions:

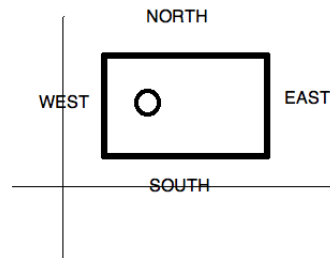
- $D_1 = (c_1 \text{ and } c_2) \text{ or } c_3$
- $D_2 = (c_1 \text{ or } c_2) \text{ and } c_3$

Question:

1. Pour chacune d'elle donnez un jeu de tests (ensemble de triplets de booléens) qui assure leur couverture MC/DC.

2 Couverture fonctionnelle

On observe la position d'un objet à l'intérieur d'une zone. L'objet est modélisé par un cercle de centre (x_0, y_0) et de rayon r . La zone est modélisée par un rectangle de coin inférieur gauche (x, y) , de largeur w et de hauteur h . Les coordonnées et les longueurs sont des entiers.



On définit 3 statuts caractérisant la position de l'objet vis-à-vis de la zone: OK, WARNING et ALERT. Le statut OK est une valeur booléenne. Les deux niveaux d'alarme WARNING et ALERT sont à valeur dans l'ensemble énuméré { NONE, NORTH, SOUTH, WEST, EAST }.

On pose

1. l'objet est *en zone* lorsque tous les points de sa circonférence sont à l'intérieur du rectangle.
2. l'objet est *en limite de zone* lorsqu'un des points au moins de sa circonférence est hors du rectangle et un autre à l'intérieur du rectangle.
3. l'objet est *hors zone* lorsque tous les points de sa circonférence sont hors des limites du rectangle.

Les valeurs des 3 statuts sont ainsi définies:

- si l'objet est *en zone*: OK=true, WARNING=NONE et ALERT=NONE;
- si l'objet est *en limite de zone*: OK=false, WARNING=*dir* et ALERT=NONE où *dir* indique laquelle des limites du rectangle est touchée parmi { NORTH, SOUTH, WEST, ou EAST };
- si l'objet est *hors zone*: OK=false, WARNING=NONE et ALERT=*dir* où *dir* indique laquelle des limites du rectangle a été franchie parmi { NORTH, SOUTH, WEST, ou EAST }.

Les fonctionnalités de l'observation sont implémentées par le composant `control`. Ses entrées sont: (x_0, y_0) , r et (x, y) , w , h . Ses sorties sont les 3 statuts OK, WARNING et ALERT.

Questions

1. Donnez un jeu de test qui assure la couverture fonctionnelle nominale des sorties du composant.
2. On suppose les caractéristiques du rectangle (x, y) , w , h fixées ainsi que le rayon r de l'objet. Déterminez en fonction de ces valeurs une partition du domaine de valeurs de (x_0, y_0) .
3. Choisissez des valeurs pour (x, y) , w , h et r ; donnez un jeu de tests pour (x_0, y_0) qui assure la couverture des données aux limites.
4. N'y a-t-il pas des valeurs de (x_0, y_0) pouvant mettre en défaut le composant ?

3 Couverture structurelle

La calcul d'un taux de rtraite dépend de plusieurs paramètres: l'âge du candidat; l'âge minimal de départ, l'âge maximal, le nombre de semestres effectués, le nombre de semestres requis, etc. Voici un pseudo-code très simplifié de ce calcul:

```
00 Si (age < ageMin):
01   taux := 0
02 Sinon:
03   diffAge := ageMax - age;
04   Si (0 < diffAge):
05     diffTrim := nbTrimReq - nbTrimEff;
06     Si (0 < diffTrim):
07       Si (diffAge < diffTrim):
08         diffTrim := diffAge;
09       Fsi;
10     taux := tauxMax - diffTrim*(decote(age));
11   Sinon:
12     taux := tauxMax;
13   Fsi
14 Sinon:
15   taux := tauxMax;
16 Fsi
10 Fsi.
```

Les âges (`age`, `ageMin` et `ageMax`) sont exprimés en trimestres. On pose:

- `ageMin` = 248 et `ageMax` = 268;
- `nbTrimReq` = 165

Questions:

1. Nommez et donnez les branches de ce code sous forme de listes de numéros de lignes.
2. Donnez une jeu de tests pour `age` et `nbTrimEff` qui assure la couverture des branches du code en indiquant pour chaque test la branche correspondante.
3. Dans ce code, le nombre de chemins est-il égal au nombre de branches ? Si non, donnez les chemins (sous forme de listes de numéros de lignes) qui ne n'ont pas été couverts par votre jeu de tests.