

Jeux de tests

pour vérifier un programme

1 Objectifs et principes

Un test cherche à répondre à la question «Le résultat du programme en fonctionnement est-il conforme à la spécification?».

Nous nous limiterons aux tests en «boîte noire», c'est à dire que le testeur ne connaît pas le contenu du programme. Il n'en connaît que la spécification (données en entrée, et résultat).

Un cas de test (ou jeu de test) est un ensemble de valeurs d'entrée, de résultats attendus en cohérence avec la spécification du programme. Le programme qui reçoit les valeurs d'entrée doit fournir le résultat attendu pour être conforme à la spécification.

1.1 Exemple 1

Une fonction convertit un angle exprimé en degré en un angle exprimé en radian.
Proposer un cas de test vérifiant cette fonction.

2 Partitions ou classes d'équivalence

Combien de cas de tests sont nécessaires?

Partitions ou classes d'équivalence : une portion d'un domaine d'entrée ou de sortie pour laquelle le comportement d'un composant ou système est supposé être le même.

2.1 Méthode

1. Identifier les partitions de valeurs valides et invalides en entrée
2. Identifier les partitions de valeurs valides et invalides en sorties. (s'il y en a)
3. Dériver les cas de tests à partir des partitions

Un cas de test est nécessaires pour chaque partition en entrées et en sorties.

2.2 Exemple 2

On considère la fonction native `abs`. Quelles classes d'équivalence sont à considérer? Proposer les jeux de test qui en découlent.

2.3 Application 3

Un programme convertit une durée $n \in \mathbb{N}$ exprimée en seconde en une durée exprimée par trois valeurs complémentaires : h heures, m minutes et s secondes.

Quelles sont les partitions nécessaires? Proposer les cas de tests en conséquences.

2.4 Application 4

Un programme évalue la qualité d'un tir cadré au foot. On note x et y les coordonnées du ballon. La distance séparant l'intérieur des deux poteaux est de 7,32 m et le bord inférieur de la barre transversale se situe à 2,44 m du sol. Pour que le tir soit considéré comme cadré, il faut $x \in [0; 7,32]$ et $y \in [0; 2,44]$. Sinon, le programme affiche « Out ».



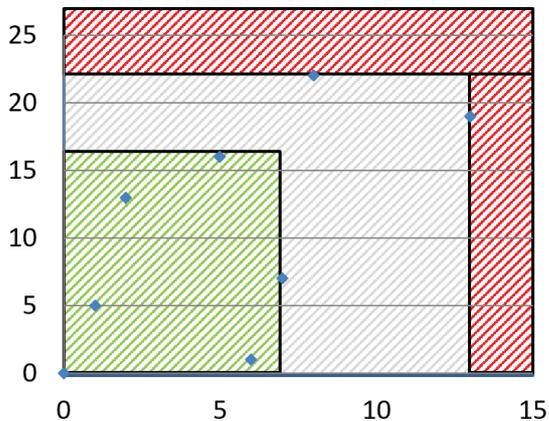
Plus le ballon est proche d'un coin, plus le tir est de qualité. Le programme calcule donc les distances entre le ballon et chaque coin. Il affiche la valeur de la distance minimale en guise de score. Plus le score est bas, meilleur est le tir.

Quelles sont les partitions nécessaires? Proposer les cas de tests en conséquences.

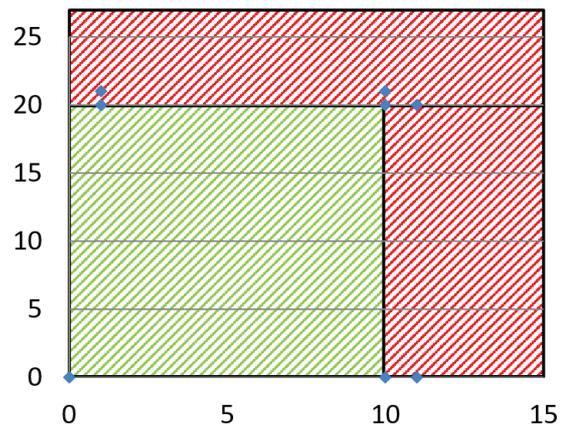
3 Tests aux limites

Combien de tests par partition sont nécessaires? On fait l'hypothèse qu'entre deux tests concluants, on peut interpoler ce résultat pour tous les cas entre ces deux premiers.

Des tests aléatoires :



Des tests basés sur les limites des partitions



3.1 Application 5

Sur la base de l'application 2.4, proposer les tests aux limites cohérents avec les partitions.