Homework 4 Group 10
Nathan Daily daily043
Wyatt Kormick kormi001
Ross Faber faber047

Problem 1

(1) Test Goals
   (a) Horizontal Green, Horizontal Walk
   (b) Vertical Green, No Walk, Horizontal Red, No Walk
   (c) Vertical Green, Walk, Horizontal Red, No Walk
   (d) Horizontal Green, No Walk → Horizontal Green, Horizontal Walk
   (e) Horizontal Yellow, No Walk → Vertical Green, No Walk
   (f) Vertical Green, Walk → Vertical Yellow, No Walk

(2) Trap Properties
   (a) AG!(mc.mode = HORIGREEN & mc.walkmode = HORIWALK);
   (b) AG!(mc.mode = VERTGREEN & mc.walkmode = NOWALK);
   (c) AG!(mc.mode = VERTGREEN & mc.walkmode = VERTWALK);
   (d) AG!(mc.mode = HORIGREEN & mc.walkmode = NOWALK & EX(mc.mode = HORIGREEN & mc.walkmode = HORIWALK));
   (e) AG!(mc.mode = HORIYELLOW & mc.walkmode = NOWALK & EX(mc.mode = VERTGREEN & mc.walkmode = NOWALK));
   (f) AG!(mc.mode = VERTGREEN & mc.walkmode = VERTWALK & EX(mc.mode = VERTYELLOW & mc.walkmode = NOWALK));

(3) NuSMV Output
   Model in file: traffic.smv
   NuSMV output in file: nusmv_output.txt

Problem 2

Test Case Explanations as Code Comments
Assume not testing Main

Faults Found

Inventory.useIngredients() adds Coffee instead of uses it
   Found in test: CoffeeMakerTest.testMakeCoffeeNotEnoughIngredients()
   Fix: change += to -=

Inventory.addSugar() adds sugar when the amount to add is negative
   Found in test: InventoryTest.testAddSugarNonpositive()
   Fix: change <= to >=

RecipeBook.addRecipe() searches for null positions in the array, but deleteRecipe() replaces with a new recipe object
   Found in test: RecipeBookTest.testDeleteRecipe()

Fix: in RecipeBook.deleteRecipe()
Replace “= new Recipe()” with “= null”

RecipeBook.editRecipe() changes the name of the new recipe to the empty string for no real reason
Found in test: RecipeBookTest.testEditRecipe()
Fix: remove line: newRecipe.setName(“”);

Test Cases in files:
CoffeeMakerTest.java
InventoryTest.java
RecipeTest.java
RecipeBookTest.java
Put them in the same folder as the original CoffeeMakerTest.java, and run Eclipse’s jUnit

Problem 3

(1)

Function One: Inventory.addChocolate
Invalid:
Delete int amtChocolate = 0;
Equivalent:
Inventory.chocolate += amtChocolate; to Inventory.chocolate += abs(amtChocolate);
Useful and Nonequivalent
if (amtChocolate >= 0) to if (amtChocolate != 0)
Valid but not Useful and Nonequivalent:
Inventory.chocolate += amtChocolate; to Inventory.chocolate += 10;

Function Two: Inventory.enoughIngredients
Invalid: Delete boolean isEnough = true;
Valid but not useful:
Boolean isEnough = true; to boolean isEnough = false;
Useful and Nonequivalent::
isEnough = false; to isEnough = true; inside one of the if statements
Equivalent:
Inventory.milk to abs(Inventory.milk)

(2)

Mutants not killed:
Equivalents for both functions
Function Two: Useful and Nonequivalent

(3)

Minimal subset of tests that kill all killable mutants:
testEnoughIngredientsEnough()

testAddChocolate()
testAddChocolateNonpositive()