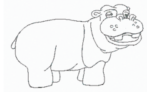
EXERCISES



4.1 Let N be the size of a sequence s of integers. Assume that an element of s can any one of v distinct values. Show that the number of possible sequences is $\Sigma_{i=0}^{N} v^i$.

4.2 An equivalence relation R on set S is reflexive, symmetric, and transitive . Also, R partitions S into equivalence classes. Show that each of the relations defined in Exercises 4.3 and 4.4 is an equivalence relation.

4.3 Derive equivalence classes for the input variables listed below.

1. `int` $pen\_inventory$; Current inventory level of writing pens.
2. `string` $planet\_name$; Planet name.
3. $operating\_system = \{$“OS X”, “Windows XP”, “Windows 2000”,
   “Unix, “Lynix”, “Xinu”, “VxWorks”$\}$;
   ; Name of an operating system.
4. `printer_class=set` printer_name; Set of printer names.
   `printer_class` p;
5. `int` $name$ [1..10]; An array of at most 10 integers.

4.4 In Example 4.4, suppose now that we add another category of printers, say “Home and home home office (hb).” Define a suitable relation hb that partitions the input domain of `pTest` into two equivalence classes. Discuss the overlap of the equivalence classes induced by hb with the remaining eight classes defined by the four relations in Example 4.4.

4.5 Consider the following relation

$cl : \mathcal{I} \rightarrow yes, no$

cl maps an input domain $\mathcal{I}$ of $pTest$ in Example 4.4 to the set $\{yes, no\}$. A printer make and model is mapped to yes if it is a color laserjet, else it is mapped to no. Is cl an equivalence relation ?

4.6 (a) Why consider classes E2-E6 in Example 4.5 when the correctness of the program corresponding to tests in these classes can be verified by simple inspection ? Offer at least two reasons. (b) Are there any additonal equivalance classes that one ought to consider while partitioning the input domain of `wordCount` ?

4.7 Partition the input domain of the $transcript$ component described in Example 4.6 into equivalence classes using the guidelines in Tables 4.1 and 4.2. Note that $transcript$ takes two inputs, a record R and an integer N.

4.8 (a) Generate two sets of tests T_1 and T_2 from the partitions created in Example 4.7 using, respectively, uni-dimensional and multidimensional testing. Which of the following relations holds amongst T_1 and T_2 that you have created: $T_1 = T_2$, $T_1 \subset T_2$, $T_1 \subseteq T_2$, $T_1 \supset T_2$, $T_1 \supseteq T_2$, and $T_1 \neq T2$? (b) Which of the six relations mentioned could hold between T_1 and T_2 assuming that T_1 is derived from equivalence classes constructed using uni-dimensional partitioning and T_2 using multidimensional partitioning ?

4.9 Consider an application App that takes two inputs $name$ and age where $name$ is a non-empty string containing at most 20 alphabetic characters and age is an integer that must satisfy the constraint $0 \leq age \leq 120$. The App is required to display an error message if the input value provided for age is out of range. The application truncates any name that is more than 20 characters in length and generates an error message if an empty string is supplied for $name$.

Partition the input domain using (a) uni-dimensional partitioning and (b) multidimensional partitioning. Construct two sets of test data test for App using the equivalence classes derived in (a) and in (b).

4.10 Suppose that an application has m input variables and that each variable partitions the input space into n equivalence classes. The multidimensional partitioning approach will divide the input domain into how many equivalence classes ?

4.11 An application takes two inputs x and y where $x \leq y$ and $-5 \leq y \leq 4$. (a) Partition the input domain using uni-dimensional and multidimensional partitioning. (b) Derive test sets based on the partitions created in (a).

4.12 In Example 4.8, we started out by calculating the number of equivalence classes to be 120. We did so because we did not account for the parent-child relationship between *cmd* and *tempch*. Given this relationship, how many equivalence classes should we start out with in the first step of the procedure for partitioning the input domain into equivalence classes ?

4.13 (a) Identify weaknesses, as many as you can, of the test T derived in Example 4.10.

(b) Derive a test set that covers each individual equivalence class derived for the four variables in Example 4.8 while ensuring that the semantic relations between different variables are maintained.

(c) Compare the test set derived in (b) with that in Table 4.3 in terms of their respective sizes and error detection effectiveness. If you believe that the error detection effectiveness of the test set you derived is less than that of the test set in Table 4.3, then offer an example of an error in the boiler control software that will likely be not detected by your test set but will likely be detected by the test set in Table 4.3.

4.14 An object named $compute$ takes an integer x as input. It is required to send a message to another object O_1 if $x \leq 0$ and message to object f_2 if $x > 0$. However, due to an error in $compute$, a message is sent to to O_1 when $x < 0$ and to O_2 otherwise. Under what condition(s) will the input $x = 0$ *not* reveal the error in $compute$?

4.15 For each test $t \in T$ in Example 4.12, construct one example of a fault in $textSearch$ that is guaranteed to be detected only by t. *Hint:* Avoid trivial examples !