Software Testing: Preparatory work for tutorial on group task 1

Category-Partition Testing

- **Prerequisites:** You should review section 11.2 of the Pezzè & Young.
- **Preparation:** Read the following functional specification (exercise 11.4 of Pezzè & Young).

Your task is to: *Derive test specifications using the category partition method for the following Airport connection check function.*

> **Airport connection check:** The airport connection check is part of an (imaginary) travel reservation system. It is intended to check the validity of a single connection between two flights in an itinerary. It is described here at a fairly abstract level, as it might be described in a preliminary design before concrete interfaces have been worked out.
>
> **Specification Signature:** Valid Connection (Arriving Flight: flight, Departing Flight: flight) returns Validity Code
>
> Validity Code 0 (OK) is returned if Arriving Flight and Departing Flight make a valid connection (the arriving airport of the first is the departing airport of the second) and there is sufficient time between arrival and departure according to the information in the airport database described below.
>
> Otherwise, a validity code other than 0 is returned, indicating why the connection is not valid.

Data types

Flight: A "flight" is a structure consisting of

- A unique identifying flight code, three alphabetic characters followed by up to four digits. (The flight code is not used by the Valid Connection function.)
- The originating airport code (3 characters, alphabetic)
- The scheduled departure time of the flight (in universal time)
- The destination airport code (3 characters, alphabetic)
- The scheduled arrival time at the destination airport.

Validity Code: The validity code is one of a set of integer values with the following interpretations:

- **0:** The connection is valid.
- **10:** Invalid airport code (airport code not found in database)
- **15:** Invalid connection, too short: There is insufficient time between arrival of first flight and departure of second flight.
- **16:** Invalid connection, flights do not connect. The destination airport of Arriving Flight is not the same as the originating airport of Departing Flight.
- **20:** Another error has been recognized (e.g., the input arguments may be invalid, or an unanticipated error was encountered).

Airport Database: The Valid Connection function uses an internal, in-memory table of airports which is read from a configuration file at system initialization. Each record in the table contains the following information:

- **Three-letter airport code.** This is the key of the table and can be used for lookups.
- **Airport zone.** In most cases the airport zone is a two-letter country code, e.g., "US" for the United States. However, where passage from one country to another is possible without a passport, the airport zone represents the complete zone in which passport-free travel is allowed. For example, the code "EU" represents the European countries which are treated as if they were a single country for purposes of travel.
- **Domestic connect time.** This is an integer representing the minimum number of minutes that must be allowed for a domestic connection at the airport. A connection is "domestic" if the originating and destination airports of both flights are in the same airport zone.
- **International connect time.** This is an integer representing the minimum number of minutes that must be allowed for an international connection at the airport. The number -1 indicates that international connections are not permitted at the airport. A connection is "international" if any of the originating or destination airports are in different zones.

Activity

Having considered the preceding specification you should do the following preparatory activity, writing brief notes as described and submit them to the designated tutor. The tutorial session will review this work and make suggestions on how to tackle the first group task.

1. From the specification, you should try to construct a list of ITFs for this specification. You may want to do this by each of you considering an individual list and then discussing how to make an amalgamated list for the whole group. There are probably only two or three ITFs for this function. **Write down the list of ITFs and for each justify why you think they are independently testable.** If you don't have a strong justification for independence of testability probably this is not an ITF.

2. Select an ITF to work on and:

   (a) Individually, each group member should work to identify the parameters and environment elements that are relevant to their ITF.

   (b) Individually, once you have identified them you should then work out characteristics of your parameters and environment elements.

   (c) As a group, discuss your lists and agree a merged list.

   (d) As a group, identify value classes for each of your characteristics.

   (e) **Write down your list of parameters and environment elements, their characteristics and the value classes you have chosen for them.**

3. As a group, work out how many test case specifications you have assuming no constraints between the value classes.

4. As a group:

   (a) Choose two or three test cases and describe what you think the result should be for each test specification.

   (b) Identify two test specification you do not think contribute much to the test effort and suggest how suitably chosen constraints could eliminate them from the set of test cases.

5. **Write down your estimate of the number of test case specifications, your examples with proposed results and the test case specification you think could be eliminated by suitably chosen constraints.**