

Functional Analysis – EC and BV

Apr/2023 Shmuel Gershon / Michael Stahl

Overview

In this exercise we continue our validation work on the T-Tweak web-based service.

In case you forgot, re-read the opening paragraphs of Exercise 2.

This exercise:

You can see the definition of all the APIs through a Swagger UI (“Swagger allows you to describe the structure of your APIs so that machines can read them” <https://swagger.io/docs/specification/2-0/what-is-swagger/>). You can also try using the APIs from the Swagger interface.

The following URL presents the specification of our service’s APIs: <http://t-tweak.gershon.info/docs>

You mission:

- a) Analyze the “password” API, list the equivalence classes of this API and propose test cases.
- b) Analyze the “Substring” API, list the boundary values and propose test cases.

Directions and hints:

General

You don’t have to run the tests you suggest, although running them may help you better understand the API. We have updated the code on the server and now you can call the APIs form Swagger.

Password

- 1) Work with the Rules of the password API to identify the ECs (they appear as part of the API documentation)
- 2) We require two equivalence partitioning:
 - a. Based on the input
 - b. Based on the output
- 3) Some simplifying rules:
 - a. Once the score value is zero, the code stops checking other rules
 - b. The order that rules are evaluated is:
 - i. Length

- ii. Same-letter or same-digit password string
 - iii. Illegal passwords ('password', 'root', 'admin')
 - iv. Checking for upper, lower and digits in the password string
 - c. If a password score was reduced since it was all "same letter" or "same digit", it does not get additional reduction due to the fact it is missing upper-case letters, lower-case letters or digits
- 4) To give you an idea how many ECs should be in the solution, here is a hint:
 - a. Input-based equivalence partition: The correct number of ECs is between 15 and 25
 - b. Output-based partition: between 5 and 15
 - 5) Once you defined the ECs, you need to give provide one test case for each EC
 - a. This would be a representative member of the EC. Example: if one EC is "strings of length 5", then **t#G1b** is a test case for this EC.
 - 6) A good partition means that any possible password string is a member of only one of the equivalence classes you defined.

Substring

- 1) BVs are of course dependent on the equivalent partition you use. Which means you need to do equivalence partitioning first.
- 2) The partitioning can be based on each input item by itself, but there is also some interaction between the input parameters, which result in additional boundary values
- 3) Hints:
 - Number of partitions for **string**: 1
 - Based on the allowed range of **string**
 - Number of partitions for **start**: 3
 - Based on the allowed range of **start**
 - Based on a relation between **start** and **string**
 - Based on a relation between **start** and **end**
 - Number of partitions for **end**: 3
 - Based on the allowed range of **end**
 - Based on a relation between **end** and **string**
 - Based on a relation between **end** and **start**
- 4) In this exercise, we do not require you to analyze output equivalence partitions
- 5) For each boundary, give the two test cases that provide the 2-point boundary value coverage. If the boundary does not depend on the value of another input, give some nominal value to the other inputs. If it is dependent on another input, specify that value of that input as well.

- a. Note: Some test cases are not feasible for trying out via the Swagger UI, since it blocks you from trying values out of range. You should still list these tests. If you want to run them (you don't have to as part of this exercise), such tests can actually be executed by direct http call to the server (look at the structure of the "request URL" Swagger shows for a valid call; try pasting it in a new browser window).

How to submit:

Submit the results in two .csv files, with the following headers (examples in purple, referring to the insurance example shown in class)

Filename: **EX3_password_<your ID>.csv**

API name	EC description	Input-EP / output EP?	Test case	Your ID
Password	Len(password) = 5	Input	5gwer	012345678
...	Etc.			
Password	<some EC>	output	<appropriate test case>	012345678

The description of an EC is assumed to mean: "All strings that match the following condition"

Filename: **EX3_substring_<your ID>.csv**

API name	Parameter	Partition	EC	Test Case			Your ID
				String	Start	End	
Substring	String	Range of string	Len(password) = 5	Abcde	2	4	012345678
Substring	String	Range of string	Len(password) = 6	34D353	1	5	012345678
							...
Substring	Start	Start/end relations	Start < 6	012345678

Substring	End	End/string relations	End > 3	012345678

Note that in “Partition” you are asked to list to which of the partitions hinted to in (3) above the boundary value applies. In the “EC” column, write which one of the equivalent classes for the given partition is covered by the proposed test case.

Grading

The exercise weight in the final grade is 3 points. It is not a mandatory exercise.

“password” EC analysis:

- Missing EC: -5%
- Over testing (more than a single tests for the same EC): -1% per test

“substring” BV analysis

- Incorrect/missing a boundary values: -3%
- Over testing (more than a single tests for the same BV): -1% per test

The deadline for submission is 3/May 23:55 .

Late submission penalty:

- One day: -1 point (out of the 3)
- Two days: -2 points (out of the 3)

You can't submit later than 2 days after the deadline.