BLACK BOX EXERCISES

Partitioning Technique

Ingeniería del Software DSIC-UPV Curso 2018-2019

Exercise 3

Apply the equivalence partitioning black box technique to obtain the test cases for a program which receives as an input a text file with the following columns:

- **Product-number:** a positive integer field minor to 256, with 3 digits.
- **Product-code:** an alphanumeric field with 4 characters.
- **Expiry-Month:** represent the months in which the product expires; it is a positive value of two digits (except 00).
- **Sale**: a field of only one character; it is "+" when the product is on sale, otherwise "-".

Provide the equivalence partitioning table with the input variable, valid classes, invalid classes and heuristic applied. Also, provide two tables the resulting with test cases, one for valid classes and another for the invalid ones. .

Exercise 3

* For numbering the valid classes, we combine the result provided by the heuristic as much as possible. Then, produce the valid test cases is easier

Input	Valid Classes	Invalid Classes	Heuristic
Product-number	(1) Number of 3 digits in [000,255] * See note	(2) No number (3) less 3 digits (4) more 3 digits (5) <000 (6) >255	Finite Values Range value
Product-code	(7) Alphanumeric of 4 characters * See note	(8) No alphanumeric (9) less 4 char (10) more 4 char	Boolean Finite value set
Expiry-Month	(11) Number of 2 digits in [01,99] * See note	(12) No number (13) Less 2 digits (14) More 2 digits (15) ='00'	Boolean Finite Values Boolean
Sale	(16) '+' (17) '-'	(18) Other value ('*')	Set of valid accepted values

Exercise 3

Input	Valid Classes	Invalid Classes	Heuristic
Product-number	(1) Number of 3 digits in [000,255] * See note	(2) No number (3) less 3 digits (4) more 3 digits (5) <000 (6) >255	Finite Values Range value
Product-code	(7) Alphanumeric of 4 characters * See note	(8) No alphanumeric (9) less 4 char (10) more 4 char	Boolean Finite value set
Expiry-Month	(11) Number of 2 digits in [01,99] * See note	(12) No number (13) Less 2 digits (14) More 2 digits (15) =’00’	Boolean Finite Values Boolean
Sale	(16) ‘+’ (17) ‘-’	(18) Other value (‘*’)	Set of valid accepted values

* For numbering the valid classes, we combine the result provided by the heuristic as much as possible. Then, produce the valid test cases is easier

	Valid Classes	Input
T.C. Valid C.	(1)(7)(11)(16)	Product-number=‘001’ ; Product-code=‘code’; Expiry-Month=’24’; Sale =‘+’
	(1)(7)(11)(17)	Product-number=‘001’ ; Product-code=‘code’; Expiry-Month=’24’; Sale =‘-’

Exercise 3

Input	Valid Classes	Invalid Classes	Heuristic
Product-number	(1) Number of 3 digits in [000,255] * See note	(2) No number (3) less 3 digits (4) more 3 digits (5) <000 (6) >255	Finite Values Reduced classes Range value
Product-code	(7) Alphanumeric of 4 characters * See note	(8) No alphanumeric (9) less 4 char (10) more 4 char	Boolean Reduced classes Finite value set
Expiry-Month	(11) Number of 2 digits in [01,99] * See note	(12) No number (13) Less 2 digits (14) More 2 digits (15) =’00’	Boolean Finite Values Reduced classes Boolean
Sale	(16) ‘+’ (17) ‘-’	(18) Other value (‘*’)	Set of valid accepted values

	Invalid Class	Input
T.C. Invalid C.	(**2**)(7)(11)(16)	Product-number=‘aaa’; Product-code=‘code’; Expiry-Month=’24’; Sale =‘+’
	(**3**)(7)(11)(16)	Product-number=‘00’; Product-code=‘code’; Expiry-Month=’24’; Sale =‘+’
	(**4**)(7)(11)(16)	Product-number=‘0000’; Product-code=‘code’; Expiry-Month=’24’; Sale =‘+’
	(**5**)(7)(11)(16)	Product-number=‘-01’; Product-code=‘code’; Expiry-Month=’24’; Sale =‘+’
	(**6**)(7)(11)(16)	Product-number=‘257’; Product-code=‘code’; Expiry-Month=’24’; Sale =‘+’
	(1)(**8**)(11)(16)	Product-number=‘001’ ; Product-code=‘-*!+’; Expiry-Month=’24’; Sale =‘+’
	(1)(**9**)(11)(16)	Product-number=‘001’ ; Product-code=‘cod’; Expiry-Month=’24’; Sale =‘+’
	(1)(**10**)(11)(16)	Product-number=‘001’ ; Product-code=‘codes’; Expiry-Month=’24’; Sale =‘+’
	(1)(7)(**12)**(16)	Product-number=‘001’ ; Product-code=‘code’; Expiry-Month=‘mm’; Sale =‘+’
	(1)(7)(**13**)(16)	Product-number=‘001’ ; Product-code=‘code’; Expiry-Month=‘0’; Sale =‘+’
	(1)(7)(**14**)(16)	Product-number=‘001’ ; Product-code=‘code’; Expiry-Month=‘000’; Sale =‘+’
	(1)(7)(**15**)(16)	Product-number=‘001’ ; Product-code=‘code’; Expiry-Month=‘00’; Sale =‘+’
	(1)(7)(11)(**18**)	Product-number=‘001’ ; Product-code=‘code’; Expiry-Month=’24’; Sale =‘*’

Exercise 4

Apply the equivalence partitioning black box technique to obtain the test cases for a method which generates a report according to the following inputs:

- Student Name: Which contains at least name and one surname
- Group: three characters, the first is a letters A,C or D, and the following two are two digits from 01 until 15
- Theory mark (T): positive number (10 maximum)
- Laboratory mark(L): positive number (10 maximum)
- Deliverable mark (D) : positive number (8 maximum)

In the report, the method adds the final course mark by means of the equation: $0.6*T+0.4*P+0.1*D$

Provide the equivalence partitioning table with the input variable, valid classes, invalid classes and heuristic applied. Also, provide two tables the resulting with test cases, one for valid classes and another for the invalid ones.

Exercise 4

* For numbering the valid classes, we combine the result provided by the heuristic as much as possible. Then, produce the valid test cases is easier

Input	Valid Classes	Invalid Classes	Heuristic
Name	(1) With at least 2 words	(2) Without at least 2 words	Boolean or must be
Group	(3) 3 char, with 1st char= A, C or D and the rest are a number in [01-15] * See note	(6) < 3 char (7) > 3 char (8) First letter not in {A,C,D} (9) 2nd and 3rd char No number (10) 2nd and 3rd <01 (11) 2nd and 3rd >15	Boolean Set of accepted values Boolean Range of values
Theory	(12) [0-10]	(13) <0 (14) >10	Range of values
Lab	(15) [0-10]	(16) <0 (17) >10	Range of values
Deliverable	(18) [0-8]	(19) <0 (20) >8	Range of values

Exercise 4

Input	Valid Classes	Invalid Classes	Heuristic
Name	(1) With at least 2 words	(2) Without at least 2 words	Boolean or must be
Group	(3) 3 char, with 1st char in {A, C or D} and the rest are a number in [01-15]	(6) < 3 char (7) > 3 char (8) First letter not in {A,C,D} (9) 2nd and 3rd char No number (10) 2nd and 3rd <01 (11) 2nd and 3rd >15	Finite value set Set of accepted values Boolean Range of values
Theory	(12) [0-10]	(13) <0 (14) >10	Range of values
Lab	(15) [0-10]	(16) <0 (17) >10	Range of values
Deliverable	(18) [0-8]	(19) <0 (20) >8	Range of values

T.C. Valid C.	Valid Classes	Input	Out
	(1)(3)(12)(15)(18)	Name='Joe Doe'; Group='A10'; Theory=10, Lab=10, Del=8	10.8

Exercise 4

Input	Valid Classes	Invalid Classes	Heuristic
Name	(1) With at least 2 words	(2) Without at least 2 words	Boolean or must be
Group	(3) 3 char, with 1st char in {A, C or D} and the rest are a number in [01-15]	(6) < 3 char (7) > 3 char (8) First letter not in {A,C,D} (9) 2nd and 3rd char No number (10) 2nd and 3rd <01 (11) 2nd and 3rd >15	Finite value set Reduced classes Set of accepted values Boolean Range of values
Theory	(12) [0-10]	(13) <0 (14) >10	Range of values
Lab	(15) [0-10]	(16) <0 (17) >10	Range of values
Deliverable	(18) [0-8]	(19) <0 (20) >8	Range of values

Invalid Class	Input	Out
(**2**)(3)(12)(15)(18)	Name='Joe'; Group='A10'; Theory=10, Lab=10, Del=8	--
(1)(**6**)(12)(15)(18)	Name='Joe Doe'; Group='A1'; Theory=10, Lab=10, Del=8	--
(1)(**7**)(12)(15)(18)	Name='Joe Doe'; Group='A001'; Theory=10, Lab=10, Del=8	--
(1)(**8**)(12)(15)(18)	Name='Joe Doe'; Group='B01'; Theory=10, Lab=10, Del=8	--
(1)(**9**)(12)(15)(18)	Name='Joe Doe'; Group='Ccc'; Theory=10, Lab=10, Del=8	--
(1)(**10**)(12)(15)(18)	Name='Joe Doe'; Group='C00'; Theory=10, Lab=10, Del=8	--
(1)(**11**)(12)(15)(18)	Name='Joe Doe'; Group='C16'; Theory=10, Lab=10, Del=8	
(1)(3)(**13**)(15)(18)	Name='Joe Doe'; Group='C10'; Theory=-1, Lab=10, Del=8	--
(1)(3)(**14**)(15)(18)	Name='Joe Doe'; Group='C10'; Theory=11, Lab=10, Del=8	--
(1)(3)(12)(**16**)(18)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=-1, Del=8	--
(1)(3)(12)(**17**)(18)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=12, Del=8	--
(1)(3)(12)(15)(**19**)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=8, Del=-1	
(1)(3)(12)(15)(**20**)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=8, Del=9	--