
SYSC 4101 / SYSC 5105

Input Space Partitioning—Criteria Part II

Reminder of Process

Tested function

- └─Function's parameters and environment variables
 - └─Parameter/variable characteristics
 - └─Characteristic's blocks
 - └─Values of block

Illustrating Criteria

- All Combinations
- Each Choice
- Pair-Wise
- Base Choice
- **Good practice:**
 1. List the test requirements/objectives required by the criterion
These are typically tuples (single, pairs, ...) of blocks
 2. Create test frames (combinations of blocks) to satisfy these objectives and check out the requirements you achieve as you go

Illustrating Criteria (Example)

- Input Model

Three characteristics with blocks [A,B], [1,2,3], and [x,y].

Base choices are underlined.

- All Combination

(A,1,x)	(A,1,y)	(A,2,x)	(A,2,y)	(A,3,x)	(A,3,y)
(B,1,x)	(B,1,y)	(B,2,x)	(B,2,y)	(B,3,x)	(B,3,y)

- Each Choice

(A,1,x)	(B,2,y)	(B,3,y)
---------	---------	---------

- Pair-Wise

(A,1,x)	(A,1,y)	(A,2,x)	(A,2,y)	(A,3,x)	(A,3,y)
(B,1,x)	(B,2,y)	(B,3,-)			

- Base Choice

(A,1,x)	(B,1,x)	(A,2,x)	(A,3,x)	(A,1,y)
---------	---------	---------	---------	---------

Concrete Example I

Specification:

A function takes two parameters:

- a string of characters
- a character

The function returns the position in the string at which the character was first found or a message indicating that the character was not present in the string.

Step 2—Parameters and Environment Variables

- There are two parameters
 - A string, let's call it S
 - A character, let's call it C
- No foreseeable environment variable

Step 3,4,5—Characteristics and Blocks

- Characteristics for parameter s :
 - Length
- Blocks for length of s
 - Empty (0)
 - Maximum (10)
 - Nominal (5)
- Characteristics for parameter c :
 - Location of (first occurrence of) c in s
 - Number of occurrences
- Blocks for location of c in s
 - Beginning of string
 - End of string
 - Middle of string (i.e., different from the other two)
- Number of occurrences
 - No occurrence
 - One occurrence
 - Several occurrences (say 2)

Beware: Values for blocks are only for illustration purposes.
This does not mean we have already chosen values.

Condensed Representation

- Length of s
 - B1 Empty
 - B2 Maximum
 - B3 Nominal
- Location of (first occurrence of) c in s
 - B4 Beginning of string
 - B5 End of string
 - B6 Middle of string
- Number of occurrences
 - B7 No occurrence
 - B8 One occurrence
 - B9 Several occurrences

Blocks have been numbered
to facilitate referencing

Adding Constraints

- Length of s

- B1 Empty
- B2 Maximum
- B3 Nominal

Means we will only exercise each block once in the set of test frames.
E.g., no need to try an empty string more than once.

Consider this a tagging of the block. Consider this defines Boolean variable `nominalLength`, which is true if we use this block in a test frame.

[error]

[single]

[property nominalLength]

- Location of (first occurrence of) c in s

- B4 Beginning of string
- B5 End of string
- B6 Middle of string

[if nominalLength and !noChar]

[if nominalLength and !noChar]

[if nominalLength and !noChar]

Only use this block if `nominalLength` is true and `noChar` is false, i.e., nominal length of string and character appears at least once

- Number of occurrences

- B7 No occurrence
- B8 One occurrence
- B9 Several occurrences

[property noChar]

[single]

Test Frames—Each Block

- B1
 - B2-B7
 - B9-B3-B4
 - B3-B4-B8
 - B3-B5-B8
 - B3-B7
 - B3-B6-B8
- Each choice appears at least once in the set of test frames
 - Greyed out block:
 - optional, to make things more complete
 - Order of blocks in a test frame irrelevant
 - Can obtain several sets of test frames
 - As long as we satisfy the criteria, all is right

Test Cases—Each Block

- | | | |
|------------|--|--------------------------|
| • B1 | Empty S, c does not matter | s="", c=? |
| • B2-B7 | Maxi length, no occurrence | s="abcdefgh...ij", c='9' |
| • B9-B3-B4 | Several occurrences, nominal length, first occurrence at index 1 | s="abcadaae", c='a' |
| • B3-B4-B8 | Nominal length, first occurrence at index 1, one occurrence | s="abcbe", c='a' |
| • B3-B5-B8 | Nominal length, first occurrence at end of string, one occurrence | s="abcde", c='e' |
| • B3-B7 | Nominal length, no occurrence | s="abcde", c='f' |
| • B3-B6-B8 | Nominal length, first occurrence in middle of string, one occurrence | s="abcde", c='c' |

Comments

- Search of several occurrences only performed with first occurrence at index 1
 - Question: Is this sufficient?
 - Answer: It depends!
 - What is important is that you do not forget something obvious and that blocks satisfy their properties (completeness, disjointness)
- We do not really control how the search of several occurrences block (B9) is combined with others.

Test Frames—Pairwise

Colours indicate which test requirements (pairs) are exercised by which test frame.

Test requirements

B1-B4	B1-B5	B1-B6
B2-B4	B2-B5	B2-B6
B3-B4	B3-B5	B3-B6
B1-B7	B1-B8	B1-B9
B2-B7	B2-B8	B2-B9
B3-B7	B3-B8	B3-B9
B4-B7	B4-B8	B4-B9
B5-B7	B5-B8	B5-B9
B6-B7	B6-B8	B6-B9

Some pairs are ~~unfeasible~~, given the semantics of blocks

(e.g., B5-B9: several occurrences and first occurrence is last character in string)

Recall the revised definition of adequacy?

Test frames

- B2-B4-B8
- ~~B2-B5-B8~~
- ~~B2-B6-B9~~
- B3-B4-B9
- B3-B5-B8
- B3-B6-B8
- B2-B7
- B3-B7
- B1

B1 is added last as we want to exercise each choice at least once with pairwise.

~~Accounting for single/error.~~

Adding Constraints—Other solution

- Length of s
 - B1 Empty [error]
 - B2 Maximum
 - B3 Nominal
- Location of (first occurrence of) c in s
 - B4 Beginning of string [!noChar]
 - B5 End of string [!noChar and !several]
 - B6 Middle of string [!noChar]
- Number of occurrences
 - B7 No occurrence [property noChar]
 - B8 One occurrence
 - B9 Several occurrences [property several]

Test Cases—Each Block

• B1	Empty S, c does not matter	s="", c=?
• B2-B4-B8	Maxi length, index 1, 1 occurrence	s="9abcdefgh...ij", c='9'
• B2-B4-B9	Maxi length, index 1, several occ.	s="9abc9defg9h...ij", c='9'
• B2-B5-B8	Maxi length, last index, 1 occ.	s="abcdefgh...ij9", c='9'
• B2-B6-B8	Maxi length, middle index, 1 occ.	s="abcd9efgh...ij", c='9'
• B2-B6-B9	Maxi length, middle index, several occ.	s="abc9de9fgh...ij9", c='9'
• B3-B4-B8	Nominal length, index 1, 1 occ.	s="abcde", c='a'
• B3-B4-B9	Nominal length, index 1, several occ.	s="abcadae", c='a'
• B3-B5-B8	Nominal length, last index, 1 occ.	s="abcde", c='e'
• B3-B6-B8	Nominal length, middle index, 1 occ.	s="abcde", c='c'
• B3-B6-B9	Nominal length, middle index, several occ.	s="abcdce", c='c'
• B3-B7	Nominal length, no occurrence	s="abcde", c='f'

Comment

- We test all possible search (several occurrences, all indexes) for both possible lengths of the string (maximum length and nominal length)
 - Question: Isn't this too much?
 - Answer: it depends!
 - What is important is that you do not forget something obvious and that blocks satisfy their properties (completeness, disjointness)

Example II

- Program P reads an input file F containing two integers m and n, and outputs the value of $1/(m+n)$.
- Parameters? Environment variables? Characteristics? Blocks? Values? Constraints? Test frame? Test cases?

Example III

Command: find

Syntax: find <pattern> <file>

Function:

- The find command is used to locate one or more instances of a given pattern in a text file. All lines in the file that contain the pattern are written to standard output. A line containing the pattern is written only once, regardless of the number of times the pattern occurs in it.
- The pattern is any sequence of characters whose length does not exceed the maximum length of a line in the file. To include a blank in the pattern, the entire pattern must be enclosed in quotes (“”). To include a quotation mark in the pattern, two quotes in a row (“”) must be used.