CS 3250
Zoha Husain (zh2yn) and Eza Rasheed (er6qt)
Assignment 5

A

1.

a) "Location of element in list" fails the disjointness property. Give an example that illustrates this.

The disjointness property says that blocks must not overlap. Disjointness property states that no element of domain is in more than one block.

Example 1:

List with 3 repeated elements:
ArrayList = [1,1,1]
element = 1
Block 1: First entry in list = 1
Block 2: Last entry in list = 1
Block 3: Some other position in list = 1

The "location of element in list" fails this property because an example list could fall under multiple blocks. For example, if the same element is repeated multiple times in an array list [1, 1, 1] and we were searching for 1, the input would fall in block 1, 2, and 3. There are many ways to demonstrate that this fails the disjointness property.

Example 2:

List with only 1 element:
ArrayList = [6]
element = 6
Block 1: First entry in list = 6
Block 2: Last entry in list = 6

If there is only one element in the array [6] and we're searching for 6, this is both the first and last element in the array (block 1 and block 2). Therefore, since 6 is in 2 blocks, it does not satisfy the disjointness property.

b) "Location of element in list" fails the completeness property. Give an example that illustrates this.

The completeness property says that the blocks must cover the entire domain. The three blocks don't describe if there is an input where the element being searched is not in the array. Thus, these three blocks don't cover all possibilities.

Example 1:

List with 3 elements:
ArrayList = [2, 4]
element = 3
Does not fall/is not included in any block

c) **Supply one or more new partitions that capture the intent of "Location of element in list" but do not suffer from completeness or disjointness problems.**
- "

To make sure that the new partitions do not suffer from completeness or disjointness, check whether these properties are true or false.

Block 1:
- If element is the first entry in the list → [true | false]

Block 2:
- If element is the last entry in the list → [true | false]

2.

2.1 List all of the input variables, including the state variables

1 textbox, 1 test requirement button (truth table), 5 test buttons (GACC, CACC, RACC, GICC, RICC), 4 other buttons (New Expression, Graph Coverage, Data Flow Coverage, Minimal - MUMCUT coverage), browser features (back, forward, refresh)

2.2 Define characteristics of the input variables. Make sure you cover all input variables
- C1: Textbox (P =) → valid string or not
- C2: Test Requirement button (Truth Table) → click or not
- C3: 5 Test Buttons (GACC, CACC, RACC, GICC, RICC) → which button
- C4: 4 Other buttons (New Expression, Graph Coverage, Data Flow Coverage, Minimal - MUMCUT coverage) → which button
- C5: Browser back → click or not

2.3 Partition the characteristics into blocks
- C1: Input to P textbox is a valid string or not [valid string | invalid string]
- C2: Truth Table is clicked or not [click | not click]
- C3: Which Logic Test Type button is clicked [GACC | CACC | RACC | GICC | RICC | None]
- C4: Which other button is clicked [New Expression | Graph Coverage | Data Flow Coverage | Minimal - MUMCUT coverage | None]
- C5: Browser back [click back | not click]

CS 3250
Zoha Husain (zh2yn) and Eza Rasheed (er6qt)
Assignment 5

2.4 Designate one block in each partition as the "Base" block

- C1: Input to P textbox is a valid logical statement or not
  - Base: Input is valid string
- C2: Truth Table is clicked or not
  - Base: Truth Table button is clicked
- C3: Which Logic Test Type button is clicked
  - Base: GACC
- C4: Which other button is clicked
  - Base: “New Expression” is clicked
- C5: Browser back is clicked or not
  - Base: Browser back is not clicked

2.5 Define values for each block

- C1: Input to P textbox is a valid logical statement or not
  - [Valid: “q ^ p”] [Invalid: “67”]
- C2: Truth Table is clicked
  - [click | not click]
- C3: Which Logic Test Type button is clicked.
  - [GACC | CACC | RACC | GICC | RICC | None]
- C4: Which other button is clicked
  - [New Expression | Graph Coverage | Data Flow Coverage | Minimal - MUMCUT coverage | None]
- C5: Browser back is clicked
  - [click back | not click back]

2.6 Write a test set (a set of test cases) that satisfies Base Choice Coverage (BCC)
Write your tests with the values from the previous step
Be sure to include inputs (test input values) and expected outputs

- TC1 (Base) = [q ^ p, Truth Table is clicked, GACC, None, browser back is not clicked]
  - Expected:

The following result for GACC is based on the truth table on the right:

Major Clause	Set of possible tests
q	(1,3), (1,4), (2,3), (2,4)
p	(1,2), (1,4), (3,2), (3,4)

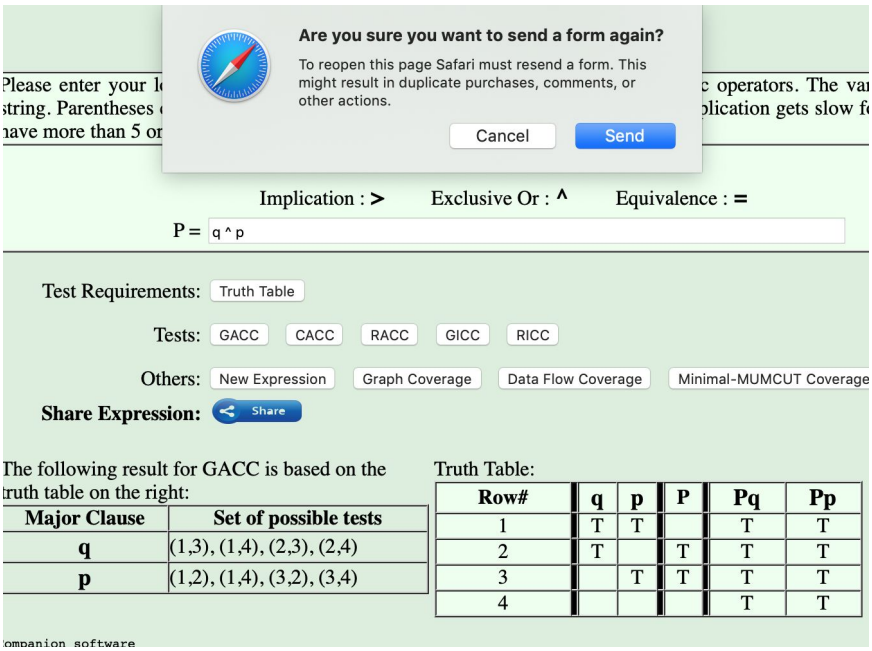
Truth Table:

Row#	q	p	P	Pq	Pp
1	T	T		T	T
2	T		T	T	T
3		T	T	T	T
4				T	T

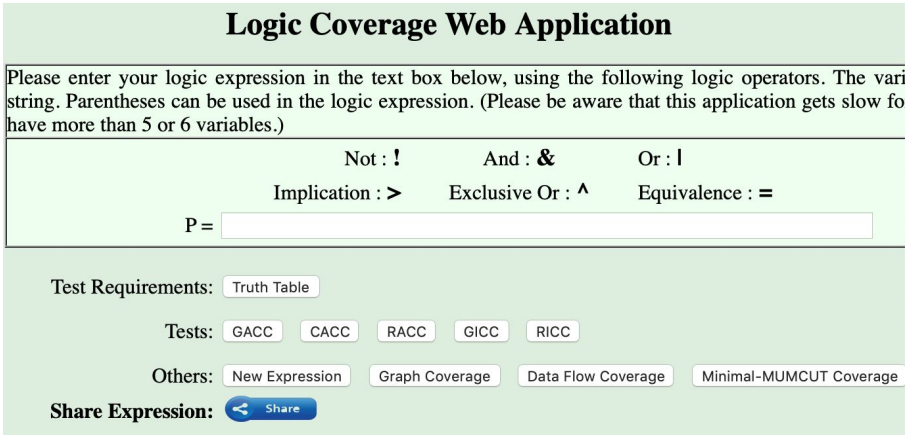
Number of Test Cases: 1 + (2-1) + (2-1) + (6-1) + (5-1) + (2-1) = 13

- TC2: [q ^ p, Truth Table is clicked, GACC, None**, browser back is clicked**]

- Expected:

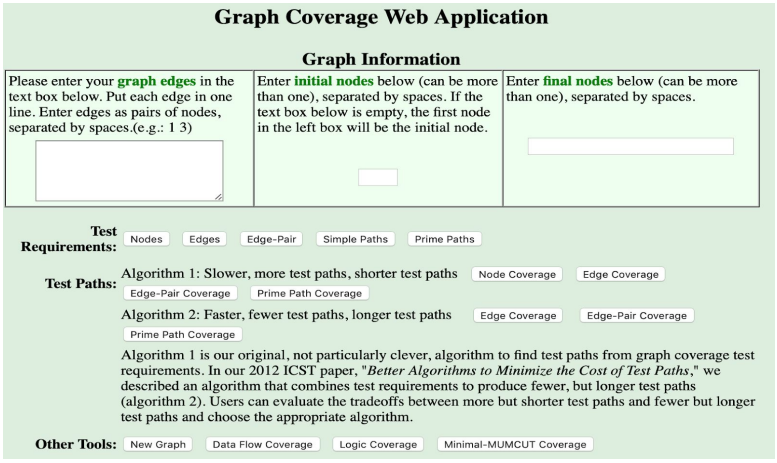


- TC3: [q ^ p, Truth Table is clicked, GACC, **New Expression clicked,** browser back is not clicked]
  - Expected:

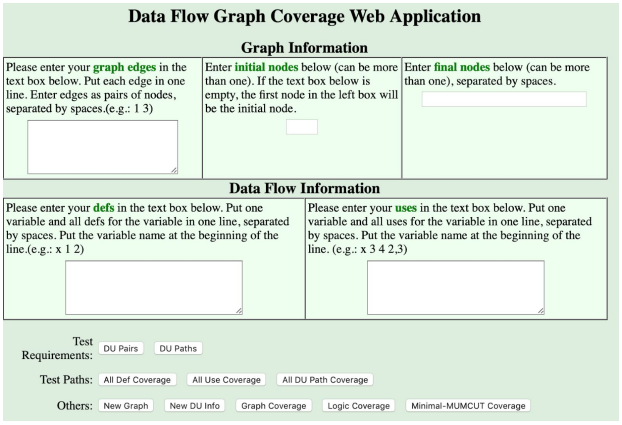


  - There is no truth table or set of possible tests
- TC4: [q ^ p, Truth Table is clicked, GACC, **Graph Coverage clicked,** browser back is not clicked]
  - Expected:

CS 3250
Zoha Husain (zh2yn) and Eza Rasheed (er6qt)
Assignment 5

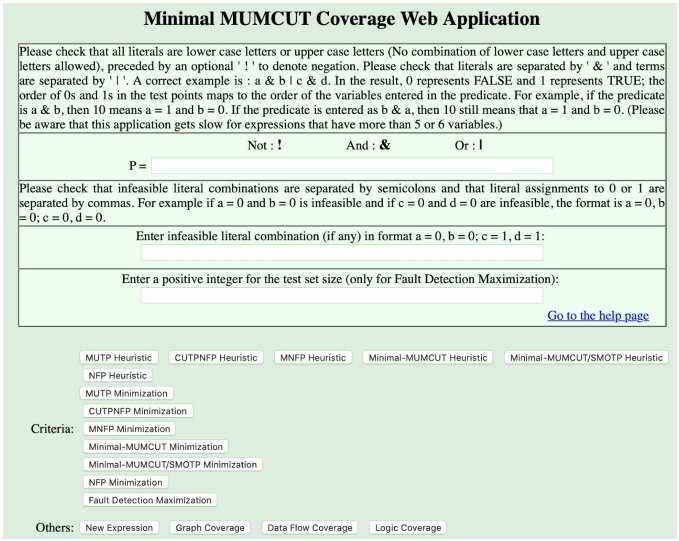


- TC5: [q ^ p, Truth Table is clicked, GACC, **Data flow coverage clicked,** browser back is not clicked]
    - Expected:



- TC6: [q ^ p, Truth Table is clicked, GACC, **Minimal - MUMCUT coverage clicked,** browser back is not clicked]
    - Expected:

CS 3250
Zoha Husain (zh2yn) and Eza Rasheed (er6qt)
Assignment 5



- TC7: [q ^ p, Truth Table is clicked, **CACC**, None, browser back is not clicked]
  - Expected:

The following result for CACC is based on the truth table on the right:

Major Clause	Set of possible tests
q	(1,3), (2,4)
p	(1,2), (3,4)

Truth Table:

Row#	q	p	P	Pq	Pp
1	T	T		T	T
2	T		T	T	T
3		T	T	T	T
4				T	T

- TC8: [q ^ p, Truth Table is clicked, **RACC**, None, browser back is not clicked]
  - Expected:

The following result for RACC is based on the truth table on the right:

Major Clause	Set of possible tests
q	(1,3), (2,4)
p	(1,2), (3,4)

Truth Table:

Row#	q	p	P	Pq	Pp
1	T	T		T	T
2	T		T	T	T
3		T	T	T	T
4				T	T

- TC9: [q ^ p, Truth Table is clicked, **GICC**, None, browser back is not clicked]
  - Expected:

The following result for GICC is based on the truth table on the right:

Major Clause	Set of possible tests

Truth Table:

Row#	q	p	P	Pq	Pp
1	T	T		T	T
2	T		T	T	T
3		T	T	T	T
4				T	T

CS 3250
Zoha Husain (zh2yn) and Eza Rasheed (er6qt)
Assignment 5

- TC10: [q ^ p, Truth Table is clicked, **RICC**, None, browser back is not clicked]
  - Expected:

The following result for RICC is based on the truth table on the right:

Major Clause	Set of possible tests

Truth Table:

Row#	q	p	P	Pq	Pp
1	T	T		T	T
2	T		T	T	T
3		T	T	T	T
4				T	T

- TC11: [q ^ p, Truth Table is clicked, **None**, None, browser back is not clicked]
  - Expected:

Truth Table:

Row#	q	p	P	Pq	Pp
1	T	T		T	T
2	T		T	T	T
3		T	T	T	T
4				T	T

- TC12: [q ^ p, **Truth Table is not clicked**, GACC, None, browser back is not clicked]
  - Expected:

The following result for GACC is based on the truth table on the right:

Major Clause	Set of possible tests
q	(1,3), (1,4), (2,3), (2,4)
p	(1,2), (1,4), (3,2), (3,4)

Truth Table:

Row#	q	p	P	Pq	Pp
1	T	T		T	T
2	T		T	T	T
3		T	T	T	T
4				T	T

  - Although the truth table was not clicked, when we clicked the GACC button, it still appeared
- TC13: [67, Truth Table is clicked, GACC, None, browser back is not clicked]
  - Expected:

unexpected char: '6'

  - Since the input was invalid, we received the response above