Lab Report

By: Jahmil Ally (501045419)

# Observations and Analysis

## Q1:

### Truth Table for Clauses in 𝑝

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | c | ¬b | ¬b ∨ c | p = a ∧ (¬b ∨ c) |
| F | F | F | T | T | F |
| F | F | T | T | T | F |
| F | T | F | F | F | F |
| F | T | T | F | T | F |
| T | F | F | T | T | T |
| T | F | T | T | T | T |
| T | T | F | F | F | F |
| T | T | T | F | T | T |

### Conditions for Clause Determination

A clause determines p when flipping the clause value changes the value of p, while keeping other variables constant.

* Clause a:
  + a determines p when ¬b∨c is true, Since p=a∧(¬b∨c).
  + Key rows: When ¬b∨c=1 p changes as a changes.
  + Conditions: a determines p when ¬b∨c=1.
* Clause b:
  + b appears as ¬b, so its effect is analyzed in ¬b∨c.
  + Key rows: When a=1 and c=0, changing b affects p.
  + Conditions: b determines p when a=1,c=0.
* Clause c:
  + c directly affects ¬b∨c influencing p.
  + Key rows: When a=1 and b=1, changing c affects p.
  + Conditions: c determines p when a = 1, b = 1.

Output:

A screenshot of a computer

AI-generated content may be incorrect.A computer screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white paper with black text

AI-generated content may be incorrect.

A screenshot of a test

AI-generated content may be incorrect.

A table with numbers and text

AI-generated content may be incorrect.

A table with black text

AI-generated content may be incorrect.

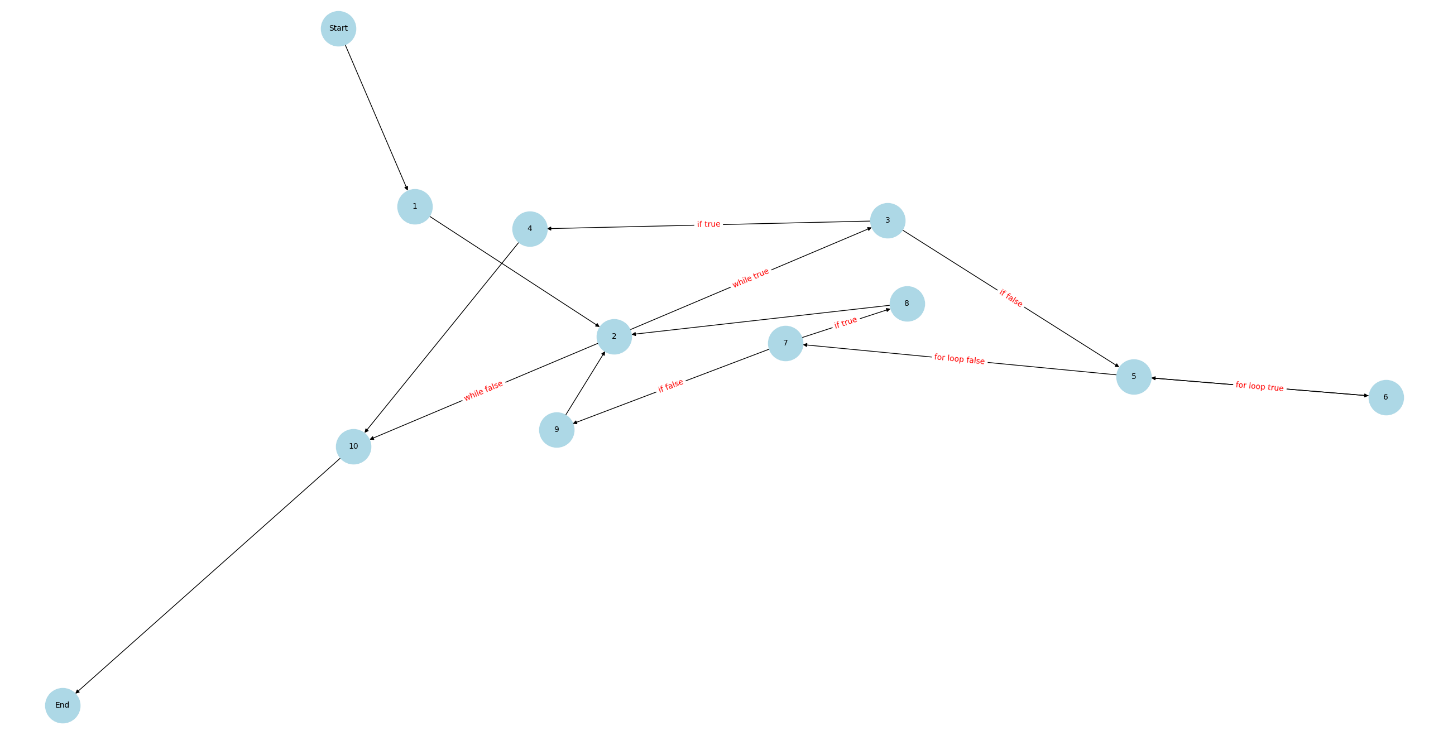
A screenshot of a table

AI-generated content may be incorrect.

## Q2:

### Data Flow Graph:

|  |  |
| --- | --- |
| Node | Lines |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4, 5 |
| 5 | 6 |
| 6 | 7 |
| 7 | 8 |
| 8 | 9 |
| 9 | 10,11 |
| 10 | 12 |



Def(n)/ Use(n)

|  |  |  |
| --- | --- | --- |
| Node | Def(n) | Use(n) |
| Start | y |  |
| 1 | x | y |
| 2 |  | x |
| 3 |  | x, y |
| 4 | x | x |
| 5 | z | x, z |
| 6 | x | x, z |
| 7 |  | x |
| 8 | y | y |
| 9 | y | y |
| 10 |  | x, y |
| End |  |  |

DU Pairs:

|  |  |  |
| --- | --- | --- |
| Variables | Defined | Used |
| X | 1, 4, 6 | 2, 3, 4, 5, 6, 7, 10 |
| Y | Start,8, 9 | 1, 3, 8, 9, 10 |
| Z | 5 | 5, 6 |

No infeasible paths

Output:

A black background with white text

AI-generated content may be incorrect.

## Q3:

### Reachability Predicates

|  |  |  |
| --- | --- | --- |
| Predicate | Condition | Effect on Reachability |
| P1 | (s1 <= 0) or (s2 <= 0) or (s3 <= 0) | If True, return INVALID, function exits. |
| P2 | (s1 + s2 <= s3) or (s2 + s3 <= s1) or (s1 + s3 <= s2) | If True, return INVALID, function exits. |
| P3 | s1 == s2 == s3 | If True, return EQUILATERAL, function exits. |
| P4 | (s1 == s2) or (s2 == s3) or (s1 == s3) | If True, return ISOSCELES, function exits. |
| P5 | Default case: None of the above predicates are true (scalene triangle). | True: All sides are different. |

### Test Requirements (PC)

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Input (s1, s2, s3) | Expected Output | Predicate Coverage Satisfied |
| TC1 | -80, 2, 3 | Triangle.INVALID | P1 = True, P2 = False |
| TC2 | 100, 2, 3 | Triangle.INVALID | P1 = False, P2 = True |
| TC3 | 3, 3, 3 | Triangle.EQUILATERAL | P1 = False, P2 = False, P3 = True |
| TC4 | 3, 3, 5 | Triangle.ISOSCELES | P1 = False, P2 = False, P3 = False, P4 = True |
| TC5 | 3, 4, 5 | Triangle.SCALENE | P1 = False, P2 = False, P3 = False, P4 = False, P5 = True |

### Test Requirements (CC)

Predicates and Atomic Conditions:

1. P1: Non-positive sides:
   1. Atomic Conditions:
      * C1.1: s1 <= 0
      * C1.2: s2 <= 0
      * C1.3: s3 <= 0
   2. Predicate:
      * C1 = C1.1 OR C1.2 OR C1.3
2. P2: Triangle inequality
   1. Atomic Conditions:
      * C2.1: s1 + s2 <= s3
      * C2.2: s2 + s3 <= s1
      * C2.3: s1 + s3 <= s2
   2. Predicate:
      * C2 = C2.1 OR C2.2 OR C2.3
3. P3: Equilateral triangle
   1. Atomic Conditions:
      * C3.1: s1 == s2
      * C3.2: s2 == s3
   2. Predicate:
      * C3 = C3.1 AND C3.2
4. P4: Isosceles triangle
   1. Atomic Conditions:
      * C4.1: s1 == s2
      * C4.2: s2 == s3
      * C4.3: s1 == s3
   2. Predicate: C4 = C4.1 OR C4.2 OR C4.3
5. P5: Scalene
   1. Predicate: !C1 OR !C2 OR!C3 OR !C4

|  |  |  |
| --- | --- | --- |
| Predicate | Atomic Conditions | TRs (True/False for each condition) |
| P1 | C1.1, C1.2, C1.3 | {[True, False, False], [False, True, False], [False, False, True]} |
| P2 | C2.1, C2.2, C2.3 | {[True, False, False], [False, True, False], [False, False, True]} |
| P3 | C3.1, C3.2 | {[True, True] , [True, False], [False, True], [False, False]} |
| P4 | C4.1, C4.2, C4.3 | {[True, False, False], [False, True, False], [False, False, True]} |
| P5 | C1, C2, C3, C4 | {[True, False, False, False], [False, True, False, False], [False, False, True, False], [False, False, False, True]} |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Input (s1, s2, s3) | Expected Output | Conditions Satisfied |
| TC1 | 0, 2, 3 | Triangle.INVALID | C1.1 = True, C1.2 = False, C1.3 = False |
| TC2 | 1, 0, 3 | Triangle.INVALID | C1.1 = False, C1.2 = True, C1.3 = False |
| TC3 | 1, 2, 0 | Triangle.INVALID | C1.1 = False, C1.2 = False, C1.3 = True |
| TC4 | 1, 2, 3 | Triangle.INVALID | C2.1 = True, C2.2 = False, C2.3 = False |
| TC5 | 3, 1, 2 | Triangle.INVALID | C2.1 = False, C2.2 = True, C2.3 = False |
| TC6 | 2, 3, 1 | Triangle.INVALID | C2.1 = False, C2.2 = False, C2.3 = True |
| TC7 | 3, 3, 3 | Triangle.EQUILATERAL | C3.1 = True, C3.2 = True |
| TC8 | 3, 3, 5 | Triangle.ISOSCELES | C4.1 = True, C4.2 = False, C4.3 = False |
| TC9 | 3, 5, 3 | Triangle.ISOSCELES | C4.1 = False, C4.2 = True, C4.3 = False |
| TC10 | 5, 3, 3 | Triangle.ISOSCELES | C4.1 = False, C4.2 = False, C4.3 = True |
| TC11 | 3, 4, 5 | Triangle.SCALENE | All conditions False |

### Determination predicates

|  |  |  |
| --- | --- | --- |
| Predicate | Simplified Expression | Meaning |
| P1 | C1 | At least one side is non-positive. |
| P2 | C2 | Triangle inequality is violated. |
| P3 | C3 | All three sides are equal (equilateral triangle). |
| P4 | C4 | At least two sides are equal (isosceles triangle). |
| P5 | !C1 OR !C2 OR !C3 OR !C4 | None of the above predicates are true (scalene triangle). |

### Test Requirements (CACC/RACC)

#### P1:

|  |  |  |  |
| --- | --- | --- | --- |
| Major Clause | Other Clauses | Predicate Outcome | Test Case Input (s1, s2, s3) |
| C1.1 | C1.2 = False, C1.3 = False | True | 0, 2, 3 |
| C1.1 | C1.2 = False, C1.3 = False | False | 1, 2, 3 |
| C1.2 | C1.1 = False, C1.3 = False | True | 1, 0, 3 |
| C1.2 | C1.1 = False, C1.3 = False | False | 1, 2, 3 |
| C1.3 | C1.1 = False, C1.2 = False | True | 1, 2, 0 |
| C1.3 | C1.1 = False, C1.2 = False | False | 1, 2, 3 |

#### P2:

|  |  |  |  |
| --- | --- | --- | --- |
| Major Clause | Other Clauses | Predicate Outcome | Test Case Input (s1, s2, s3) |
| C2.1 | C2.2 = False, C2.3 = False | True | 1, 2, 3 |
| C2.1 | C2.2 = False, C2.3 = False | False | 3, 4, 5 |
| C2.2 | C2.1 = False, C2.3 = False | True | 3, 1, 2 |
| C2.2 | C2.1 = False, C2.3 = False | False | 3, 4, 5 |
| C2.3 | C2.1 = False, C2.2 = False | True | 2, 3, 1 |
| C2.3 | C2.1 = False, C2.2 = False | False | 3, 4, 5 |

#### P3:

|  |  |  |  |
| --- | --- | --- | --- |
| Major Clause | Other Clauses | Predicate Outcome | Test Case Input (s1, s2, s3) |
| C3.1 | C3.2 = True | True | 3, 3, 3 |
| C3.1 | C3.2 = True | False | 3, 3, 5 |
| C3.2 | C3.1 = True | True | 3, 3, 3 |
| C3.2 | C3.1 = True | False | 3, 5, 3 |

#### P4:

|  |  |  |  |
| --- | --- | --- | --- |
| Major Clause | Other Clauses | Predicate Outcome | Test Case Input (s1, s2, s3) |
| C4.1 | C4.2 = False, C4.3 = False | True | 3, 3, 5 |
| C4.1 | C4.2 = False, C4.3 = False | False | 3, 4, 5 |
| C4.2 | C4.1 = False, C4.3 = False | True | 3, 5, 3 |
| C4.2 | C4.1 = False, C4.3 = False | False | 3, 4, 5 |
| C4.3 | C4.1 = False, C4.2 = False | True | 5, 3, 3 |
| C4.3 | C4.1 = False, C4.2 = False | False | 3, 4, 5 |

### Infeasible Requirements

1. P3 (Equilateral Triangle) and P4 (Isosceles Triangle): If P3 (all sides are equal) is true, then P4 (at least two sides are equal) is also true. However, the reverse is not always true. Testing for P3 being false while P4 is true is feasible, but testing for P3 being true while P4 is false is infeasible.
2. P1 (Non-positive sides) and P2 (Triangle Inequality): If P1 is true (one or more sides are non-positive), then P2 (triangle inequality) is irrelevant because the triangle is already invalid. Testing for P1 being true while P2 is false is infeasible.
3. P5 (Scalene Triangle) and P3/P4: If P5 (all sides are different) is true, then both P3 (all sides equal) and P4 (at least two sides equal) must be false. Testing for P5 being true while P3 or P4 is true is infeasible.

# Appendix:

## Q1:

## A computer screen shot of a program code AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect. A screen shot of a computer program

AI-generated content may be incorrect.

## Q2:

## A screen shot of a computer program AI-generated content may be incorrect.

## A screen shot of a computer program AI-generated content may be incorrect. A screen shot of a computer program AI-generated content may be incorrect.

## Q3:

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.