**Problem I**

1. **Branch- Testing**

Given code for the method triangle\_type () can be converted into the following control flow graph shown in below figure.

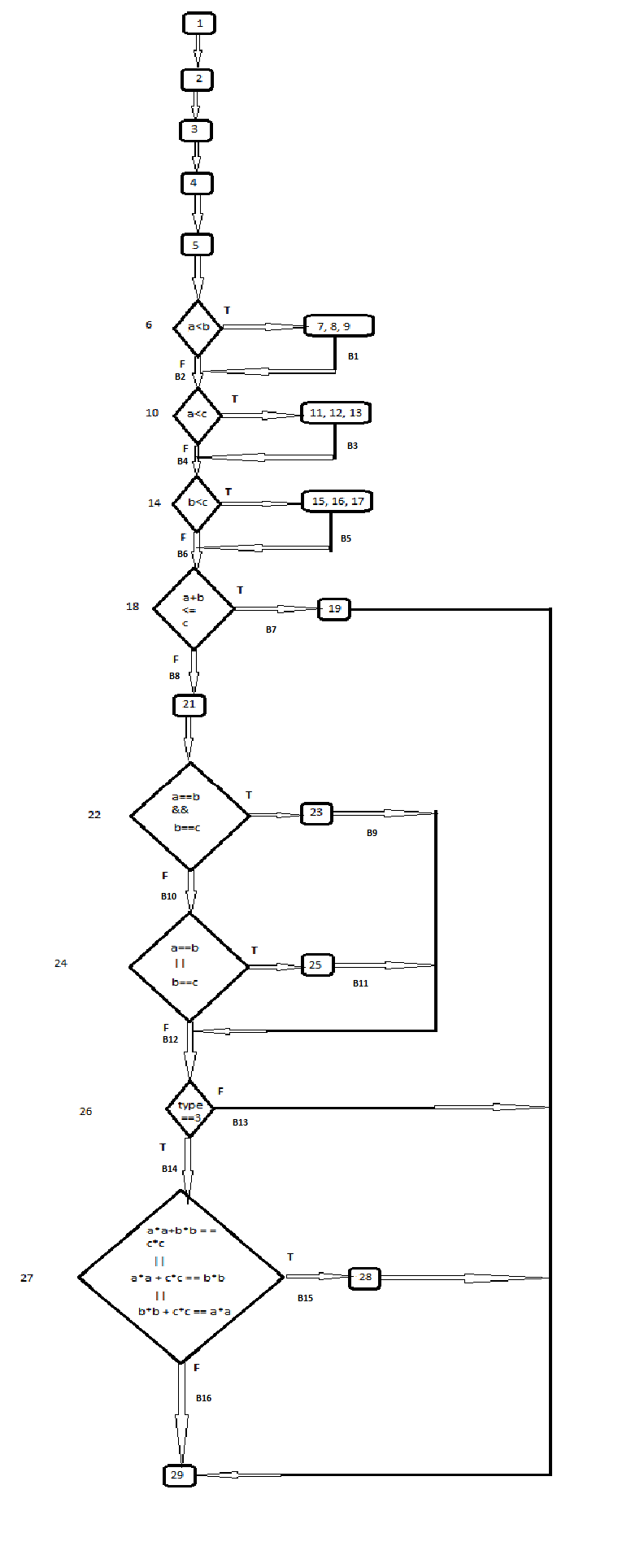
From the figure obtained the total number of branches that can be viewed are: 16

For representation purpose I have named the branches starting from B1 to B16.

Hence the test cases we design should cover each branch should get executed at least once.

The following are the test cases designed that covers all the branches from B1 till B16 shown in below figure.

|  |  |  |  |
| --- | --- | --- | --- |
| Test # | Test Case | Branches Covered  (B1- B16) | Triangle Type |
| T#1 | x1=1, y1=1, x2=1, y2=11, x3=1, y3=7 | B1, B3, B5, B7 | Not a Triangle |
| T#2 | X1=-1, y1=8.66,x2=4, y2=0, x3=-6, y3=0 | B2, B4, B6, B8, B9, B13 | Equilateral Triangle |
| T#3 | X1=0, y1=0,x2=4, y2=10, x3=8, y3=0 | B10, B11 | Isosceles Triangle |
| T#4 | X1=0, y1=0,x2=8, y2=0, x3=2, y3=6 | B12, B14, B16 | Scalene Triangle |
| T#5 | X1=0, y1=0,x2=4, y2=0, x3=0, y3=5 | B15 | Right Scalene |



1. **Multiple- Condition Testing**

From the given code for the method triangle\_type () the statements 22, 24 and 27 have complex predicates

Hence we derive multiple condition test cases for those three statements.

Statement 22:  **if (a==b && b == c)**

|  |  |  |
| --- | --- | --- |
| a == b | b == c | Test Cases |
| True | True | T#1: X1=-1, y1=8.66,x2=4, y2=0, x3=-6, y3=0 |
| True | False | T#2: X1=4, y1=10,x2=8, y2=0, x3=0, y3=0 |
| False | True | T#3: X1=0, y1=0, x2=8, y2=0, x3=4, y3=10 |
| False | False | T#4: X1=0, y1=0, x2=8, y2=0, x3=2, y3=6 |

Statement 24: **if (a == b || b == c)**

|  |  |  |
| --- | --- | --- |
| **a == b** | **b == c** | **Test Cases** |
| True | True | Test case is not possible as this case is considered as Equilateral triangle and Statement 22 will suffice this. |
| True | False | T#5: X1=4, y1=10,x2=8, y2=0, x3=0, y3=0 |
| False | True | T#6: X1=0, y1=0, x2=8, y2=0, x3=4, y3=10 |
| False | False | T#7: X1=0, y1=0, x2=8, y2=0, x3=2, y3=6 |

Statement 27: **if ((a\*a+b\*b==c\*c) || (a\*a+c\*c==b\*b) || (b\*b+c\*c==a\*a))**

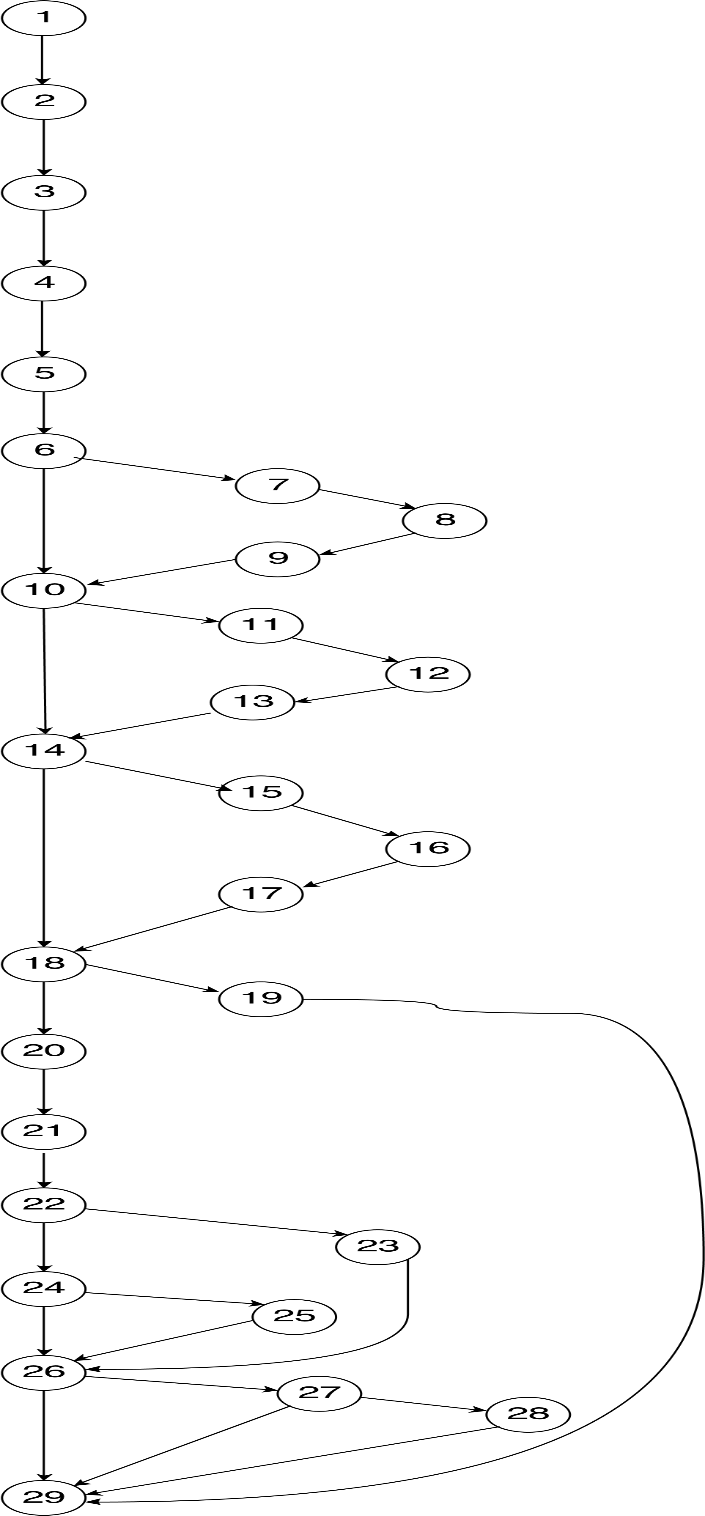
|  |  |  |  |
| --- | --- | --- | --- |
| **a\*a+b\*b==c\*c** | **a\*a+c\*c==b\*b** | **b\*b+c\*c==a\*a** | **Test Cases** |
| True | True | True | Test case not possible as no such triangle/co-ordinates with this condition exists |
| True | True | False | Test case not possible as no such triangle/co-ordinates with this condition exists |
| True | False | True | Test case not possible as no such triangle/co-ordinates with this condition exists |
| True | False | False | T#8: X1=0, y1=0,x2=4, y2=0, x3=0, y3=5 |
| False | True | True | Test case not possible as no such triangle/co-ordinates with this condition exists |
| False | True | False | T#9: X1=0, y1=0,x2=4, y2=0, x3=0, y3=5 |
| False | False | True | T#10: X1=0, y1=0,x2=4, y2=0, x3=0, y3=5 |
| False | False | False | T#11: X1=0, y1=0,x2=8, y2=0, x3=2, y3=6 |
|  |  |  |  |

**Problem II**

**Data flow Testing:**

From the given code for the method triangle\_type () the following variable are considered for the data flow testing a, b, c, type

The control flow graph drawn for this code is shown below.



**Variable “a”:**  
The following is the definition and Usage statements of variable “**a”**

|  |  |  |
| --- | --- | --- |
| **Definition** | **Usage** | **Possible Definition - Usage Pair** |
| 3 | 6 | 3->6, 3->7, 3->10, 3->11, 3->18, 3->22, 3->24, 3->27 |
| 8 | 7 | 8->10, 8->11, 8->18, 8->22, 8->24, 8->27 |
| 12 | 10 | 12->18, 12->22, 12->24, 12->27 |
|  | 11 |  |
|  | 18 |  |
|  | 22 |  |
|  | 24 |  |
|  | 27 |  |

**Variable “b”:**   
The following is the definition and Usage statements of variable “**b”**

|  |  |  |
| --- | --- | --- |
| **Definition** | **Usage** | **Possible Definition - Usage Pair** |
| 4 | 6 | 4->6, 4->8, 4->14, 4->15, 4->18, 4->22, 4->24, 4->27 |
| 9 | 8 | 9->14, 9->15, 9->18, 9->22, 9->24, 9->27 |
| 16 | 14 | 16->18, 16->22, 16->24, 16->27 |
|  | 15 |  |
|  | 18 |  |
|  | 22 |  |
|  | 24 |  |
|  | 27 |  |

**Variable “c”:**   
The following is the definition and Usage statements of variable “**c”**

|  |  |  |
| --- | --- | --- |
| **Definition** | **Usage** | **Possible Definition - Usage Pair** |
| 5 | 6 | 5->10, 5->12, 5->14, 5->16, 5->18, 5->22, 5->24, 5->27 |
| 13 | 8 | 13->14, 13->16, 13->18, 13->22, 13->24, 13->27 |
| 17 | 14 | 17->18, 17->22, 17->24, 17->27 |
|  | 15 |  |
|  | 18 |  |
|  | 22 |  |
|  | 24 |  |
|  | 27 |  |

Note: For **Variable c**: 13->18, 13->22, 13->24, 13->27 these definition – usage pairs cannot be executed as the co-ordinates satisfying this data flow are not possible

**Variable “type”:**   
The following is the definition and Usage statements of variable **“type”**

|  |  |  |
| --- | --- | --- |
| **Definition** | **Usage** | **Possible Definition - Usage Pair** |
| 2 | 26 | No definition-usage pair for definition for type in statement 2. |
| 19 | 28 | 19->29, |
| 21 | 29 | 21->26, 21->28, 21->29 |
| 23 |  | 23->26, 23-> 29 |
| 25 |  | 25->26, 25->29 |
| 28 |  | 28->29 |

**Test Cases**  
The following are the test cases for the definition-use pairs

|  |  |  |
| --- | --- | --- |
| **Test#** | **Test Case** | **Definition-Usage Pairs covered** |
| T#1: | x1=8, y1=0, x2=2, y2=6, x3=0, y3=0 | **Variable a:**  3->6, 3->7, 8->10, 8->11, 12->18, 12->22, 12->24, 12->27  **Variable b:**  4->6, 4->8, 9->14, 9->15, 16->18, 16->22, 16->24, 16->27  **Variable c:**  5->10, 5->12, 13->14, 13->16, 17->18, 17->22, 17->24, 17->27  **Variable type:** 21->26, 21->29,  **Scalene Triangle** |
| T#2 | X1=-1, y1=8.66,x2=4, y2=0, x3=-6, y3=0 | **Variable a:** 3->10, 3->18, 3->22  **Variable b:** 4->14, 4->18, 4->22  **Variable c:** 5->14, 5->18, 5->22  **Variable type:** 23->26, 23->29  **Equilateral Triangle** |
| T#3 | X1=0, y1=0,x2=4, y2=10, x3=8, y3=0 | **Variable a:** 8->18, 8->22, 8->24  **Variable b:** 9->18, 9->22, 9->24  **Variable c:** 5->24  **Variable type:** 25->26, 25->29  **Isosceles Triangle** |
| T#4 | X1=0, y1=0,x2=4, y2=0, x3=0, y3=5 | **Variable a:** 3->24, 3->27  **Variable b:** 4->24, 4->27  **Variable c:** 5->27  **Variable type:** 21->29, 28->29  **Right Scalene Triangle** |
| T#5 | X1=0, y1=0,x2=0, y2=5, x3=4, y3=0 | **Variable a:** 8->27  **Variable b:** 9->27  **Variable c:**  **Variable type:**  **Right Scalene Triangle** |
| T#6 | X1=0, y1=5,x2=0, y2=0, x3=4, y3=0 | **Variable a:** 3->11  **Variable b:**  **Variable c:**  **Variable type:**  **Right Scalene Triangle** |
| T#7 | X1=0, y1=4,x2=0, y2=0, x3=0, y3=5 | **Variable a:**  **Variable b:** 4->15  **Variable c:** 5->16  **Variable type:**  **Right Scalene Triangle** |
| T#8 | x1=1, y1=1, x2=1, y2=11, x3=1, y3=7 | **Variable a:**  **Variable b:**  **Variable c:**  **Variable type:** 19->29  **Not a Triangle** |

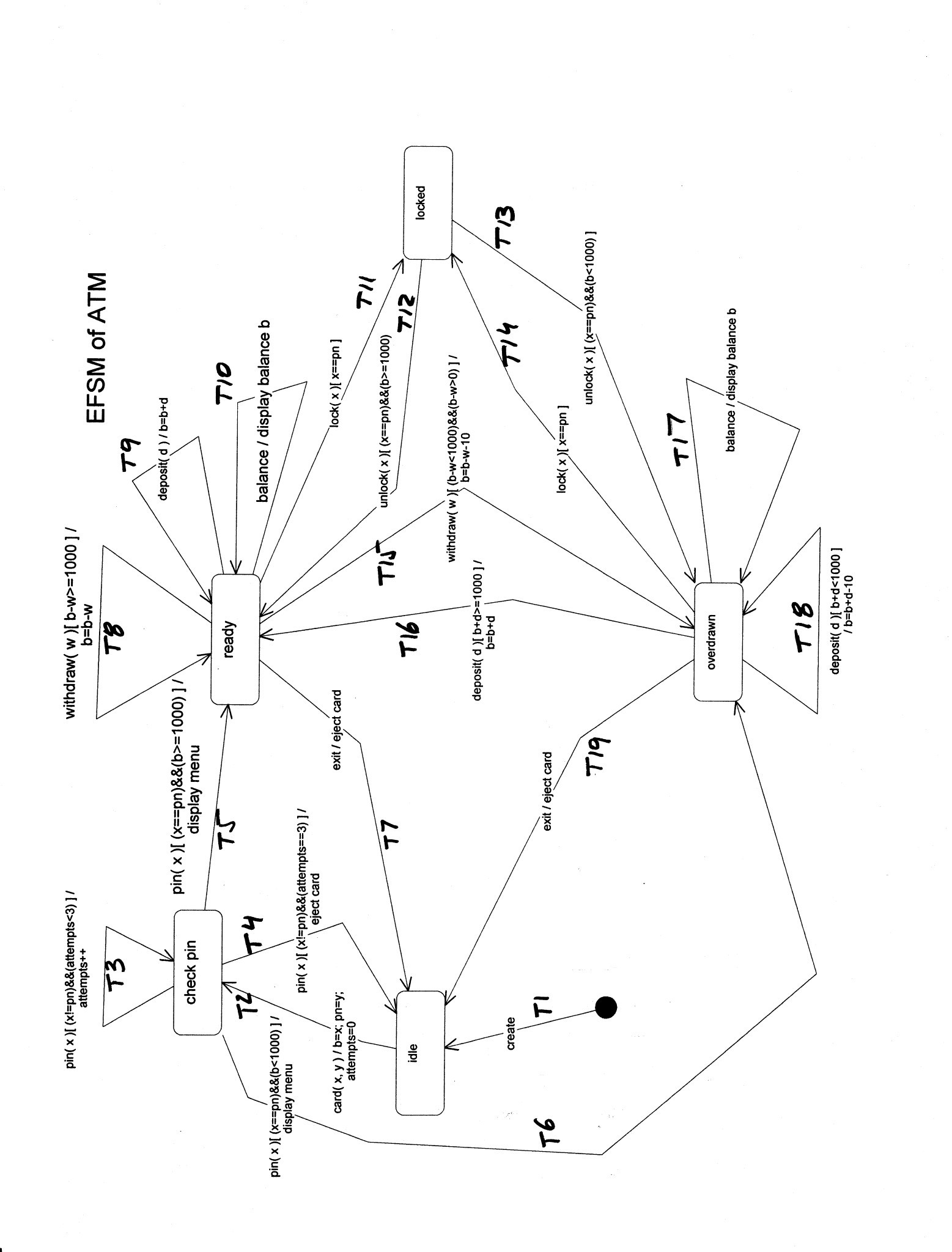
**Problem III**

1. **Transition Testing:**

From the given Extended Finite State Machine diagram the following are the transactions available

T1, T2, T3,T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, T17, T18, T19

Transaction based test cases cover such that every transaction is executed at least once



The following are the test cases designed to cover all the transactions using transaction testing.

|  |  |  |
| --- | --- | --- |
| Test# | Test Case | Transactions Covered |
| T#1 | create(), card(5000, “raghu”), pin(“qwe”), pin(“raghu”), deposit(“200”), exit() | T1, T2, T3, T5, T9, T7 |
| T#2 | create(), card(200, “raghu”), pin(“raghu”), deposit(“500”), deposit(“700”), withdraw(“1100”), exit() | T6, T18, T16, T15, T19 |
| T#3 | create(), card(5000, “raghu”), pin(“qwe”), pin(“qwe”), pin(“qwe”), deposit(“200”), exit() | T4 |
| T#4 | create(), card(5000, “raghu”), pin(“raghu”), withdraw(“200”), balance(), lock(“raghu”), unlock(“raghu”), exit() | T8, T10, T11, T12 |
| T#5 | create(), card(200, “raghu”), pin(“raghu”), lock(“raghu”), unlock(“raghu”), balance(), exit() | T13, T14, T17 |

1. **Transition – pairing Testing:**