Software Testing Lab

CS6474
Assignment-3

Shubham Laxman Hippargi

224CS3014
Master of Technology
224cs3014@nitrkl.ac.in

Department of Computer Science & Engineering
NIT, Rourkela

Selenium Assignment - 3

Shubham Laxman Hippargi
Master of Technology
224cs3014@nitrkl.ac.in

January 2025

Master of Technology
Department of Computer Science & Engineering
NIT, Rourkela

Contents

1 Find-test Testcase Reducer **3**
1.1 Find-Input Input 3
1.2 Explanation 5

2 Ice-Cream-input Testcase Reducer **6**
2.1 Ice-cream Input 6
2.2 Explanation 9

3 Annotations Testcase Reducer **10**
3.1 Annotations Input 10
3.2 Explanation 11

4 tcases-Input Testcase Reducer **13**
4.1 tcases-input Input 13
4.2 Explanation 18

1 Find-test Testcase Reducer

1.1 Find-Input Input

Below is the content of the Xml test case file:

Listing 1: External XML Code File

```
<System name="Examples">
  <Function name="find">
    <!--
         Usage: find pattern file

         Locates 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.
      -->
    <Input type="arg">
      <VarSet name="pattern" when="fileExists">
        <Var name="size">
          <Value name="empty" property="empty"/>
          <Value name="singleChar" property="singleChar"
             />
          <Value name="manyChars"/>
        </Var>
        <Var name="quoted">
          <Value name="yes" property="quoted"/>
          <Value name="no" whenNot="empty"/>
          <Value name="unterminated" failure="true"/>
        </Var>
        <Var name="blanks" whenNot="empty">
          <Value name="none"/>
```

```
31          <Value name="one" when="quoted, singleChar"/>
32          <Value name="many">
33            <When>
34              <AllOf property="quoted">
35                <Not property="singleChar"/>
36              </AllOf>
37            </When>
38          </Value>
39        </Var>
40        <Var name="embeddedQuotes" whenNot="empty,
             singleChar">
41          <Value name="none"/>
42          <Value name="one"/>
43          <Value name="many" once="true"/>
44        </Var>
45      </VarSet>
46
47      <Var name="fileName">
48        <Value name="defined" property="fileName"/>
49        <Value name="missing" failure="true"/>
50      </Var>
51    </Input>
52
53    <Input type="env">
54      <VarSet name="file" when="fileName">
55        <Var name="exists">
56          <Value name="yes" property="fileExists"/>
57          <Value name="no" failure="true"/>
58        </Var>
59        <VarSet name="contents" when="fileExists"
             whenNot="empty">
60          <Var name="linesLongerThanPattern">
61            <Value name="one" property="matchable" once=
                 "true"/>
62            <Value name="many" property="matchable"/>
63            <Value name="none" failure="true"/>
64          </Var>
65          <Var name="patterns" when="matchable" whenNot=
               "empty">
66            <Value name="none" once="true"/>
67            <Value name="one" property="match"/>
68            <Value name="many" property="match, many"/>
69          </Var>
```

```
            <Var name="patternsInLine" when="match">
              <Value name="one"/>
              <Value name="many" once="true" when="many"/>
            </Var>
          </VarSet>
        </VarSet>
      </Input>

  </Function>
</System>
```

→Test Cases Without Reducers

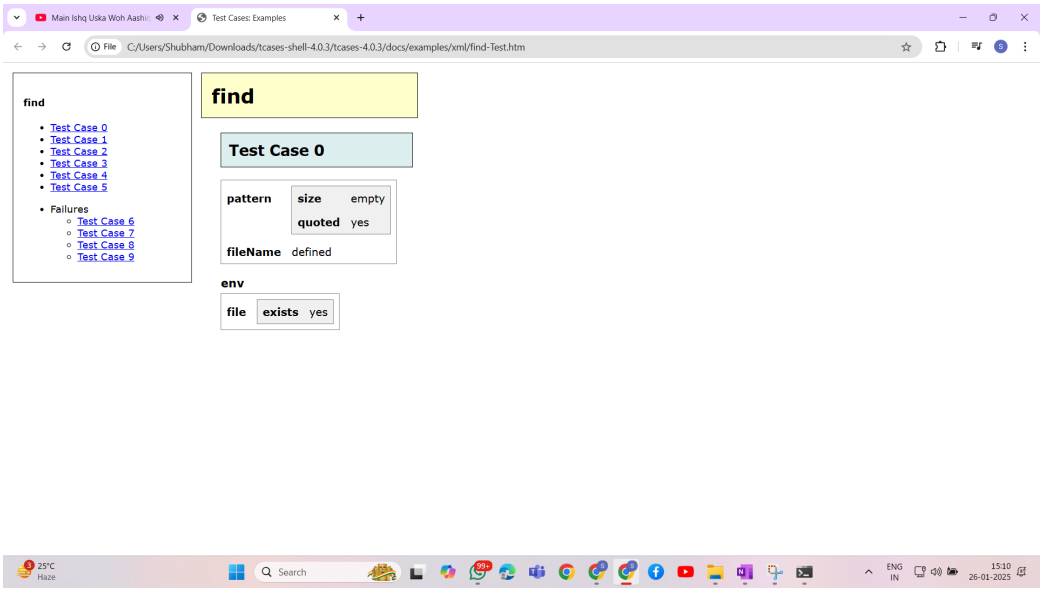

Figure 1: Test cases without reducer.

→Test Cases With Reducers

1.2 Explanation

The reducer was expected to optimize the test cases by minimizing their count. It aims to eliminate unnecessary duplicates or overlapping scenarios, ensuring a more efficient and streamlined set of test cases.However, the results show no reduction in test cases, suggesting the reducer is not functioning effectively.

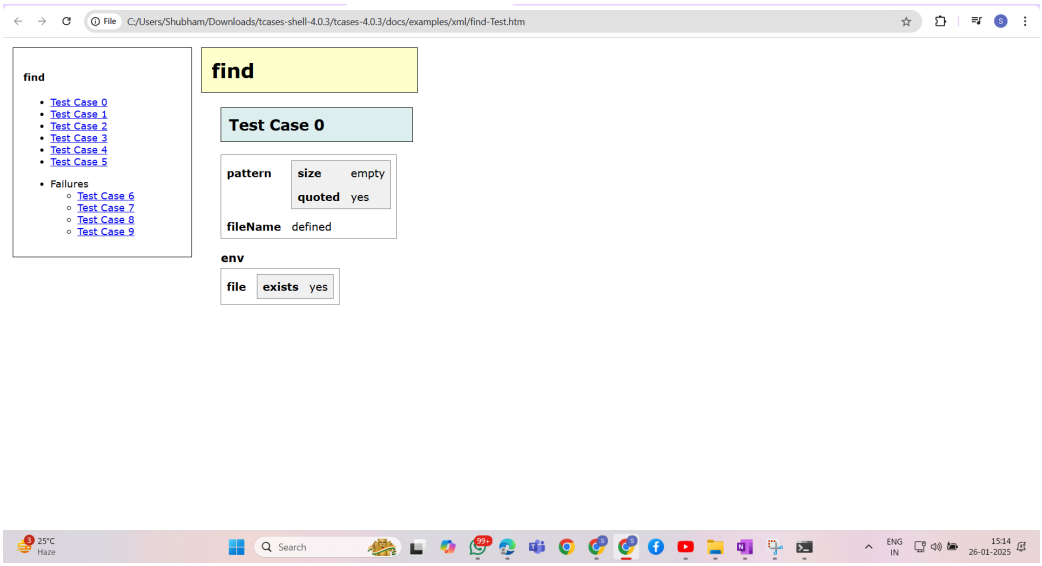

Figure 2: Test cases with reducer.

2 Ice-Cream-input Testcase Reducer

2.1 Ice-cream Input

Below is the content of the Xml test case file:

Listing 2: External XML Code File

```
<System name="Ice-Cream">
    <Function name="Cones">
        <Input>
            <Var name="Cone">
                <Value name="Empty" failure="true">
                    <When>
                        <LessThan property="scoop" max="
                           1"/>
                    </When>
                </Value>
                <Value name="Plain">
                    <When>
                        <AllOf>
                            <Equals property="scoop"
                               count="1"/>
                            <NotMoreThan property="
                               topping" max="1"/>
                        </AllOf>
```

```
            </When>
        </Value>
        <Value name="Plenty">
            <When>
                <AllOf>
                    <Between property="scoop"
                       min="1" max="2"/>
                    <NotMoreThan property="
                       topping" max="2"/>
                </AllOf>
            </When>
        </Value>
        <Value name="Grande">
            <When>
                <AllOf>
                    <Between property="scoop"
                       exclusiveMin="0"
                       exclusiveMax="4"/>
                    <Between property="topping"
                       min="1" max="3"/>
                </AllOf>
            </When>
        </Value>
        <Value name="Too-Much" failure="true">
            <When>
                <AnyOf>
                    <MoreThan property="scoop"
                       min="3"/>
                    <NotLessThan property="
                       topping" min="4"/>
                </AnyOf>
            </When>
        </Value>
    </Var>

    <VarSet name="Flavors">
        <Var name="Vanilla">
            <Value name="Yes" property="scoop"/>
            <Value name="No"/>
        </Var>
        <Var name="Chocolate">
            <Value name="Yes" property="scoop"/>
            <Value name="No"/>
```

```
            </Var>
            <Var name="Strawberry">
                <Value name="Yes" property="scoop"/>
                <Value name="No"/>
            </Var>
            <Var name="Pistachio">
                <Value name="Yes" property="scoop"/>
                <Value name="No"/>
            </Var>
            <Var name="Lemon">
                <Value name="Yes" property="scoop"/>
                <Value name="No"/>
            </Var>
            <Var name="Coffee">
                <Value name="Yes" property="scoop"/>
                <Value name="No"/>
            </Var>
        </VarSet>

        <VarSet name="Toppings" when="scoop">
            <Var name="Sprinkles">
                <Value name="Yes" property="topping"
                   />
                <Value name="No"/>
            </Var>
            <Var name="Pecans">
                <Value name="Yes" property="topping"
                   />
                <Value name="No"/>
            </Var>
            <Var name="Oreos">
                <Value name="Yes" property="topping"
                   />
                <Value name="No"/>
            </Var>
            <Var name="Cherries">
                <Value name="Yes" property="topping"
                   />
                <Value name="No"/>
            </Var>
            <Var name="MMs">
                <Value name="Yes" property="topping"
                   />
```

```
                    <Value name="No"/>
                </Var>
                <Var name="Peppermint">
                    <Value name="Yes" property="topping"
                       />
                    <Value name="No"/>
                </Var>
            </VarSet>
        </Input>
    </Function>
</System>
```

→**Test Cases Without Reducers**

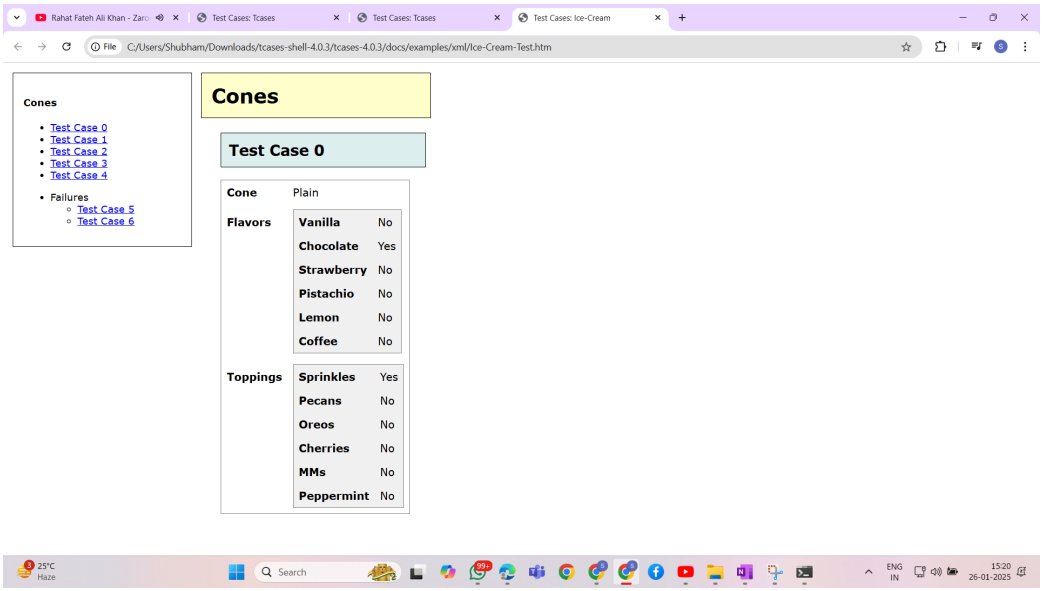

Figure 3: Test cases without reducer.

→**Test Cases With Reducers**

2.2 Explanation

The reducer was expected to optimize the test cases by minimizing their count.However, the results show no reduction in test cases, suggesting the reducer is not functioning effectively.

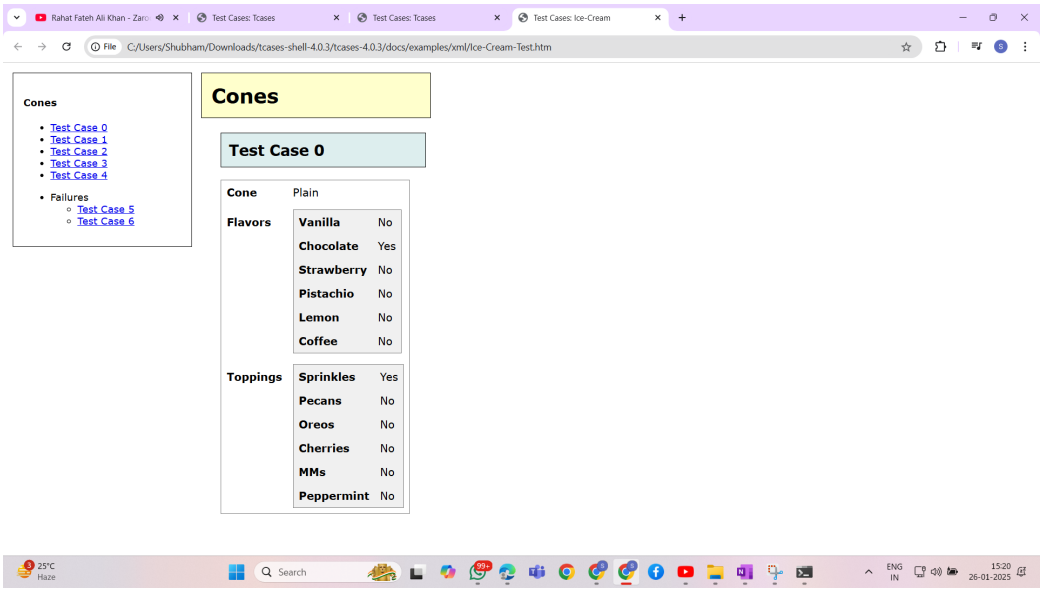

Figure 4: Test cases with reducer.

3 Annotations Testcase Reducer

3.1 Annotations Input

Below is the content of the Xml test case file:

Listing 3: External XML Code File

```
<System name="Examples">
    <Function name="addShape">
        <!-- Test case annotations -->
        <Has name="pageType" value="Page"/>
        <Has name="pageName" value="page"/>
        <Has name="pageValue" value="new Page()"/>

        <Input>

            <Var name="Type">
                <!-- Variable binding annotations -->
                <Has name="varType" value="Shape"/>
                <Has name="varName" value="shape"/>
                <Has name="varEval" value="new Shape"/>

                <Value name="SQUARE"/>
                <Value name="CIRCLE"/>
```

```
                <Value name="LINE" property="1D"/>
            </Var>

            <Var name="Size">
                <!-- Variable binding annotations -->
                <Has name="varType" value="int"/>
                <Has name="varName" value="size"/>
                <Has name="varApply" value="setSize"/>

                <Value name="1"/>
                <Value name="10"/>
                <Value name="100" property="Large"/>
            </Var>

            <Var name="Color">
                <!-- Variable binding annotations -->
                <Has name="varType" value="String"/>
                <Has name="varName" value="color"/>
                <Has name="varApply" value="setColor"/>

                <Value name="red"/>
                <Value name="green"/>
                <Value name="blue"/>
            </Var>
        </Input>
    </Function>
</System>
```

→Test Cases Without Reducers
→Test Cases With Reducers

3.2 Explanation

The reducer was expected to optimize the test cases by minimizing their count.However, the results show no reduction in test cases, suggesting the reducer is not functioning effectively.

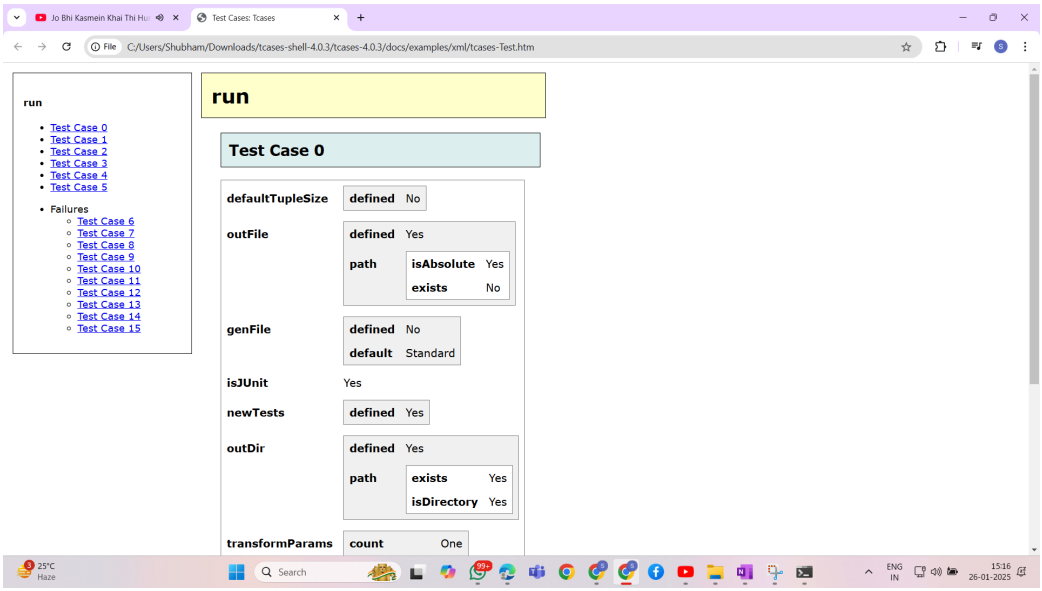

Figure 5: Test cases without reducer.

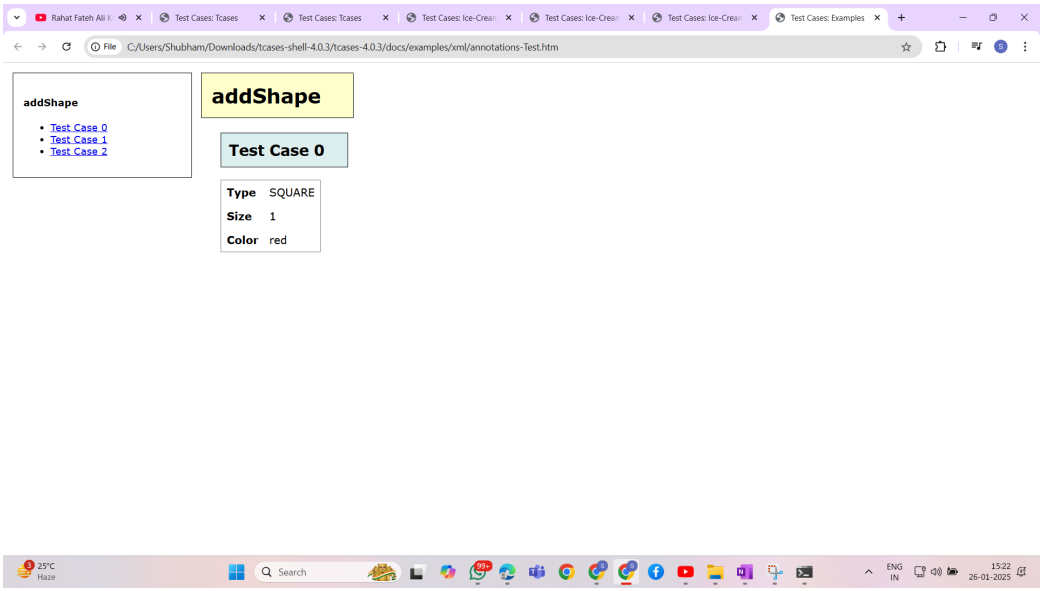

Figure 6: Test cases with reducer.

4 tcases-Input Testcase Reducer

4.1 tcases-input Input

Below is the content of the Xml test case file:

Listing 4: External XML Code File

```
<System name="Tcases">
  <Function name="run">
    <Input>

      <!-- Option: -c -->
      <VarSet name="defaultTupleSize">
        <Var name="defined">
          <Value name="Yes" property="defaultTupleSize"
             />
          <Value name="No"/>
        </Var>

        <Var name="isNumber" when="defaultTupleSize">
          <Value name="Yes"/>
          <Value name="No" failure="true"/>
        </Var>
      </VarSet>

      <!-- Option: -f -->
      <VarSet name="outFile">
        <Var name="defined">
          <Value name="Yes" property="outFile"/>
          <Value name="No"/>
          <Value name="TransformOutputUndefined" failure
             ="true" when="transformedOut,
             testFileExists"/>
        </Var>

        <VarSet name="path" when="outFile">
          <Var name="isAbsolute">
            <Value name="Yes"/>
            <Value name="No"/>
          </Var>
          <Var name="exists">
            <Value name="Yes" property="outFileExists"/>
            <Value name="No"/>
          </Var>
```

```
    </VarSet>
  </VarSet>

  <!-- Option: -g -->
  <VarSet name="genFile">
    <Var name="defined">
      <Value name="Yes" property="genFile"/>
      <Value name="No"/>
    </Var>

    <VarSet name="path" when="genFile">
      <Var name="isAbsolute">
        <Value name="Yes"/>
        <Value name="No"/>
      </Var>
      <Var name="exists">
        <Value name="Yes"/>
        <Value name="No" failure="true"/>
      </Var>
    </VarSet>

    <Var name="default" whenNot="genFile">
      <Value name="ForInputExists" when="inFile"/>
      <Value name="ForInputNone" when="inFile"/>
      <Value name="Standard" whenNot="inFile"/>
    </Var>
  </VarSet>

  <!-- Option: -J -->
  <Var name="isJUnit">
    <Value name="Yes" property="isJUnit,
       transformedOut" whenNot="transform"/>
    <Value name="No" when="transform"/>
    <Value name="NotAllowed" when="transform"
       failure="true"/>
  </Var>

  <!-- Option: -n -->
  <VarSet name="newTests">
    <Var name="defined" when="testFileExists">
      <Value name="Yes"/>
      <Value name="No"/>
    </Var>
```

```
    </VarSet>

    <!-- Option: -o -->
    <VarSet name="outDir">
      <Var name="defined">
        <Value name="Yes" property="outDir"/>
        <Value name="No"/>
      </Var>

      <VarSet name="path" when="outDir">
        <Var name="exists">
          <Value name="Yes" property="outDirExists"/>
          <Value name="No"/>
        </Var>
        <Var name="isDirectory" when="outDirExists">
          <Value name="Yes"/>
          <Value name="No" failure="true"/>
        </Var>
      </VarSet>
    </VarSet>

    <!-- Option: -p -->
    <VarSet name="transformParams">
      <When>
        <AnyOf property="transform, isJUnit"/>
      </When>
      <Var name="count">
        <Value name="One" property="params"/>
        <Value name="Many" property="params"/>
        <Value name="None"/>
      </Var>
      <Var name="assignsValue" when="params">
        <Value name="Yes"/>
        <Value name="No" failure="true"/>
      </Var>
      <Var name="nameDefined" when="params">
        <Value name="Yes"/>
        <Value name="No" failure="true"/>
      </Var>
      <Var name="valueDefined" when="params">
        <Value name="Yes"/>
        <Value name="No"/>
      </Var>
```

```
    </VarSet>

    <!-- Option: -r -->
    <VarSet name="seed">
      <Var name="defined">
        <Value name="Yes" property="random"/>
        <Value name="No"/>
      </Var>

      <Var name="isNumber" when="random">
        <Value name="Yes"/>
        <Value name="No" failure="true"/>
      </Var>
    </VarSet>

    <!-- Option: -t -->
    <VarSet name="testFile">
      <Var name="defined">
        <Value name="Yes" property="testFile"/>
        <Value name="No"/>
      </Var>

      <VarSet name="path" when="testFile">
        <Var name="isAbsolute">
          <Value name="Yes"/>
          <Value name="No"/>
        </Var>
        <Var name="exists">
          <Value name="Yes" property="testFileExists"
             />
          <Value name="No"/>
        </Var>
      </VarSet>

      <VarSet name="default" whenNot="testFile">
        <Var name="exists">
          <Value name="Yes" property="testFileExists"
             />
          <Value name="No"/>
        </Var>
      </VarSet>
    </VarSet>
```

```
160       <!-- Option: -x -->
161       <VarSet name="transform">
162         <Var name="defined">
163           <Value name="Yes" property="transform,
                transformedOut"/>
164           <Value name="No"/>
165         </Var>
166         <VarSet name="path" when="transform">
167           <Var name="isAbsolute">
168             <Value name="Yes"/>
169             <Value name="No"/>
170           </Var>
171           <Var name="exists">
172             <Value name="Yes"/>
173             <Value name="No" failure="true"/>
174           </Var>
175         </VarSet>
176       </VarSet>
177
178       <!-- Input definition file -->
179       <VarSet name="inFile">
180         <Var name="defined">
181           <Value name="Yes" property="inFile"/>
182           <Value name="No"/>
183         </Var>
184         <VarSet name="path" when="inFile">
185           <Var name="isAbsolute">
186             <Value name="Yes"/>
187             <Value name="No"/>
188           </Var>
189           <Var name="exists">
190             <Value name="asDefined"/>
191             <Value name="withInputXml"/>
192             <Value name="withXml"/>
193             <Value name="No" failure="true"/>
194           </Var>
195         </VarSet>
196       </VarSet>
197
198     </Input>
199   </Function>
200 </System>
```

→Test Cases Without Reducers

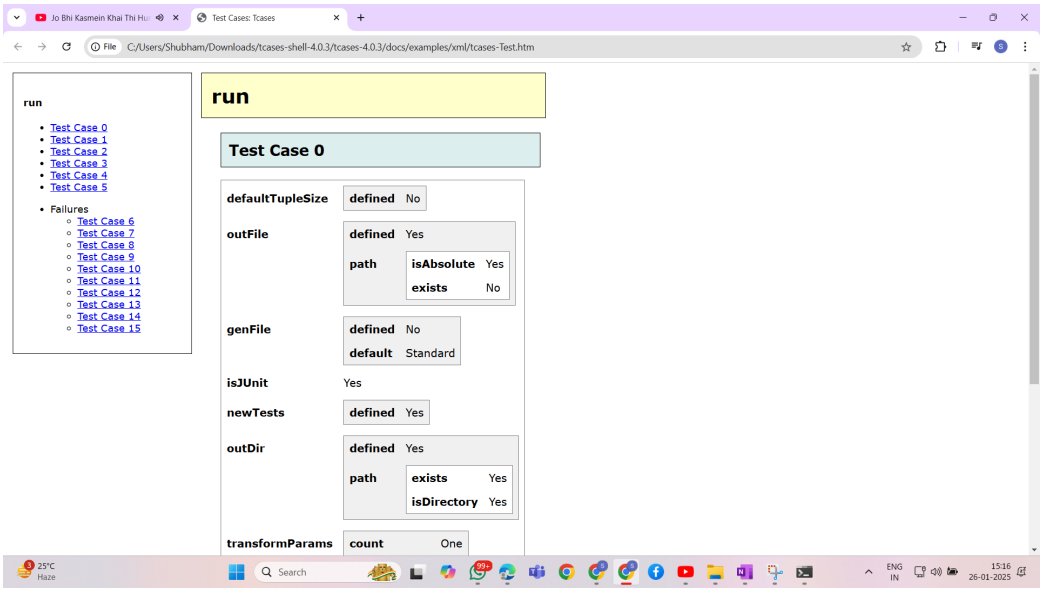

Figure 7: Test cases without reducer.

→Test Cases With Reducers

4.2 Explanation

The reducer was expected to optimize the test cases by minimizing their count.However, the results show no reduction in test cases, suggesting the reducer is not functioning effectively.

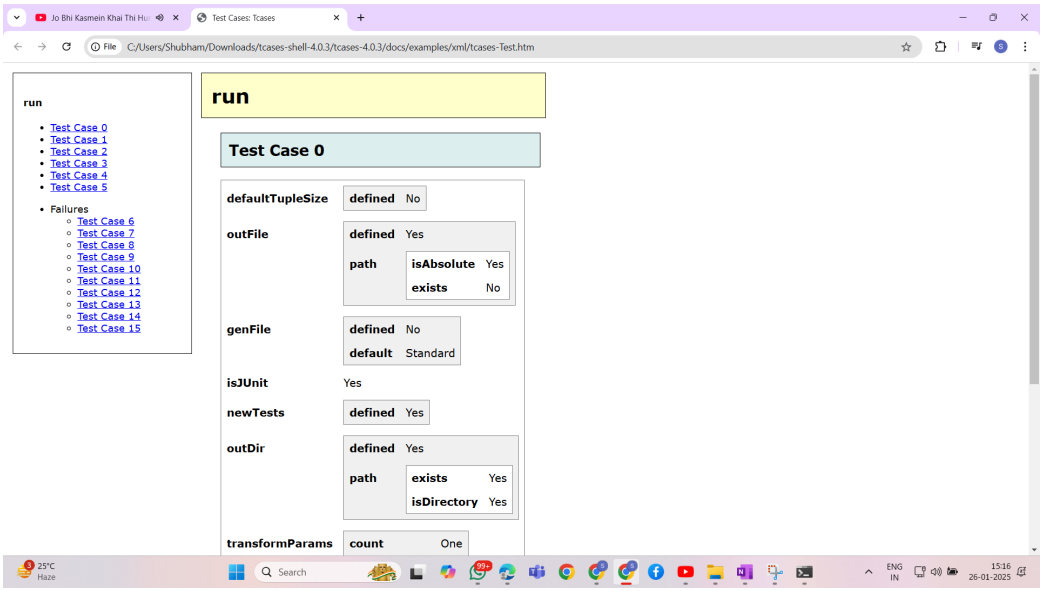

Figure 8: Test cases with reducer.