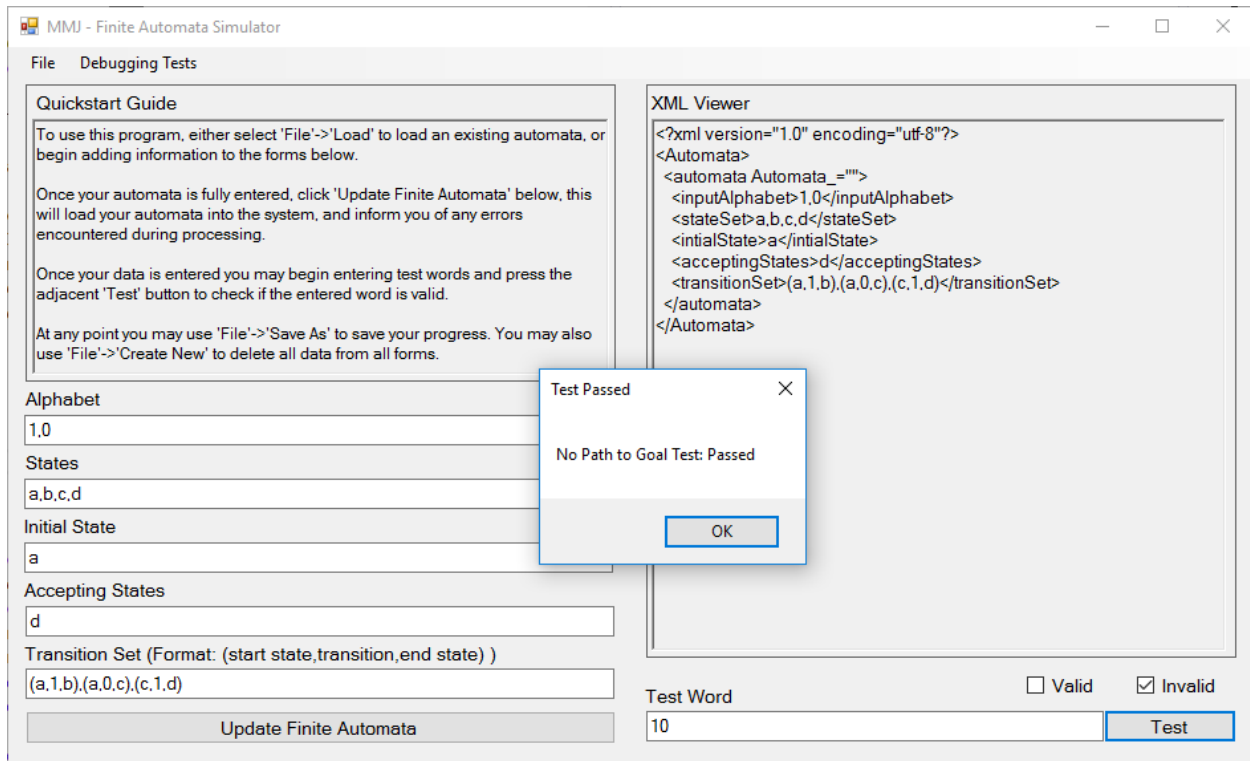


MMJ-Automation

Milestone 7

No Path to Accepting:



Expected Behavior: Invalid

This will handle to case of passing a word into an automaton object that has no way to get to an accepting state.

Circular Path:

The screenshot shows the MMJ - Finite Automata Simulator window. On the left, there is a 'Quickstart Guide' and a form for defining the automata. The form includes fields for 'Alphabet' (0,1), 'States' (a,b,c), 'Initial State' (a), 'Accepting States' (a), and 'Transition Set' ((a,0,b),(b,0,c),(c,1,a),(c,1,b)). An 'Update Finite Automata' button is at the bottom of the form. On the right, an 'XML Viewer' displays the XML representation of the automata. A 'Test Passed' dialog box is centered over the interface, displaying 'Circular Path Test: Passed' and an 'OK' button. At the bottom right, there is a 'Test Word' field containing '001', with 'Valid' and 'Invalid' checkboxes (the 'Valid' checkbox is checked) and a 'Test' button.

MMJ - Finite Automata Simulator

File Debugging Tests

Quickstart Guide

To use this program, either select 'File'-'>'Load' to load an existing automata, or begin adding information to the forms below.

Once your automata is fully entered, click 'Update Finite Automata' below, this will load your automata into the system, and inform you of any errors encountered during processing.

Once your data is entered you may begin entering test words and press the adjacent 'Test' button to check if the entered word is valid.

At any point you may use 'File'-'>'Save As' to save your progress. You may also use 'File'-'>'Create New' to delete all data from all forms.

Alphabet

0,1

States

a,b,c

Initial State

a

Accepting States

a

Transition Set (Format: (start state,transition,end state))

(a,0,b),(b,0,c),(c,1,a),(c,1,b)

Update Finite Automata

XML Viewer

```
<?xml version="1.0" encoding="utf-8"?>
<Automata>
  <automata Automata_="">
    <inputAlphabet>0,1</inputAlphabet>
    <stateSet>a,b,c</stateSet>
    <initialState>a</initialState>
    <acceptingStates>a</acceptingStates>
    <transitionSet>(a,0,b),(b,0,c),(c,1,a),(c,1,b)</transitionSet>
  </automata>
</Automata>
```

Test Passed

Circular Path Test: Passed

OK

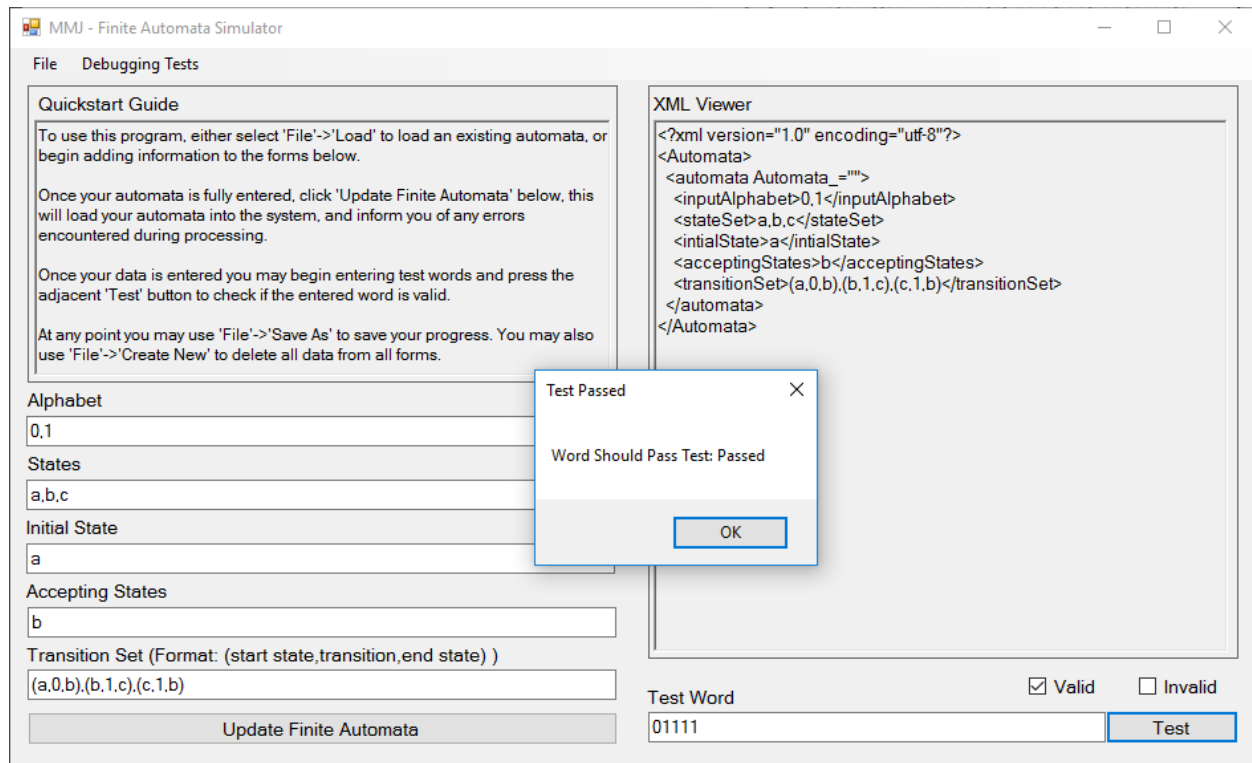
Test Word ☒ Valid ☐ Invalid

001

Expected Behavior: Valid

This test will handle the case of passing a word into an automaton object that contains a loop within the states of the automaton.

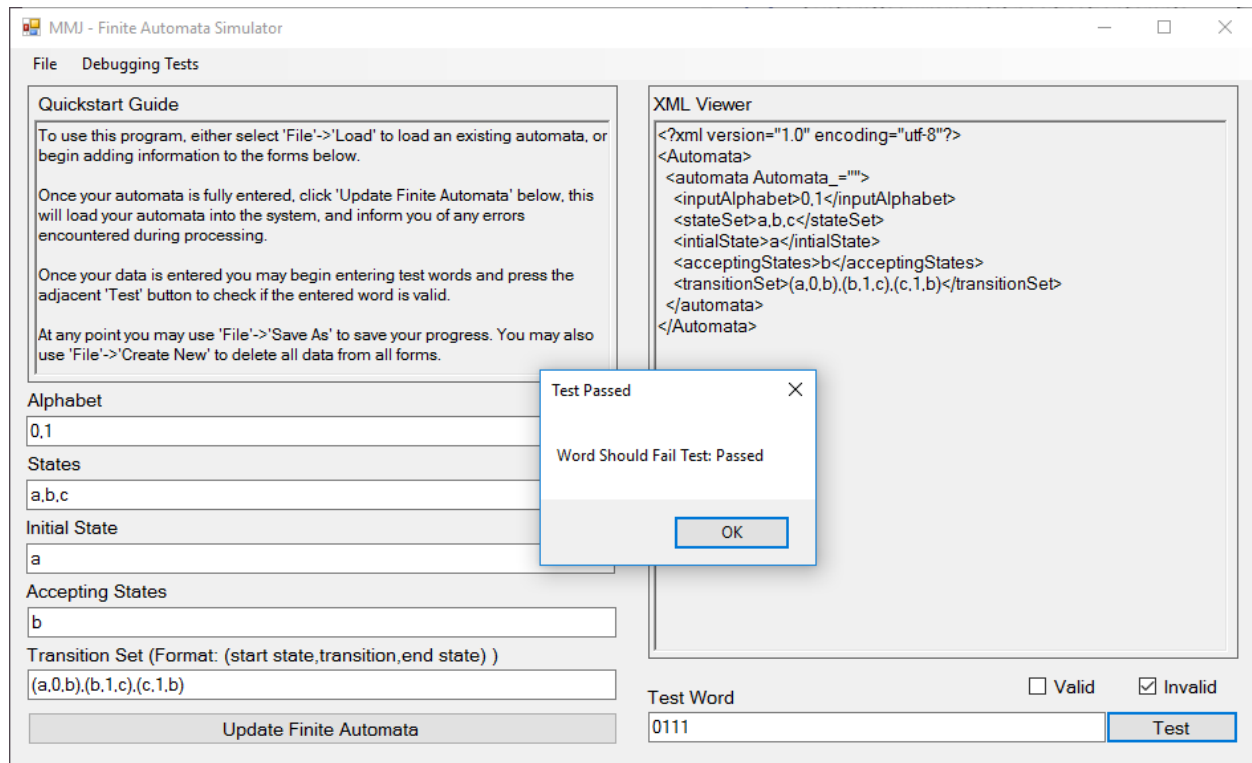
Standard Success:



Expected Behavior: Valid

This test will handle the case of passing in a word that should be accepted by an automaton object.

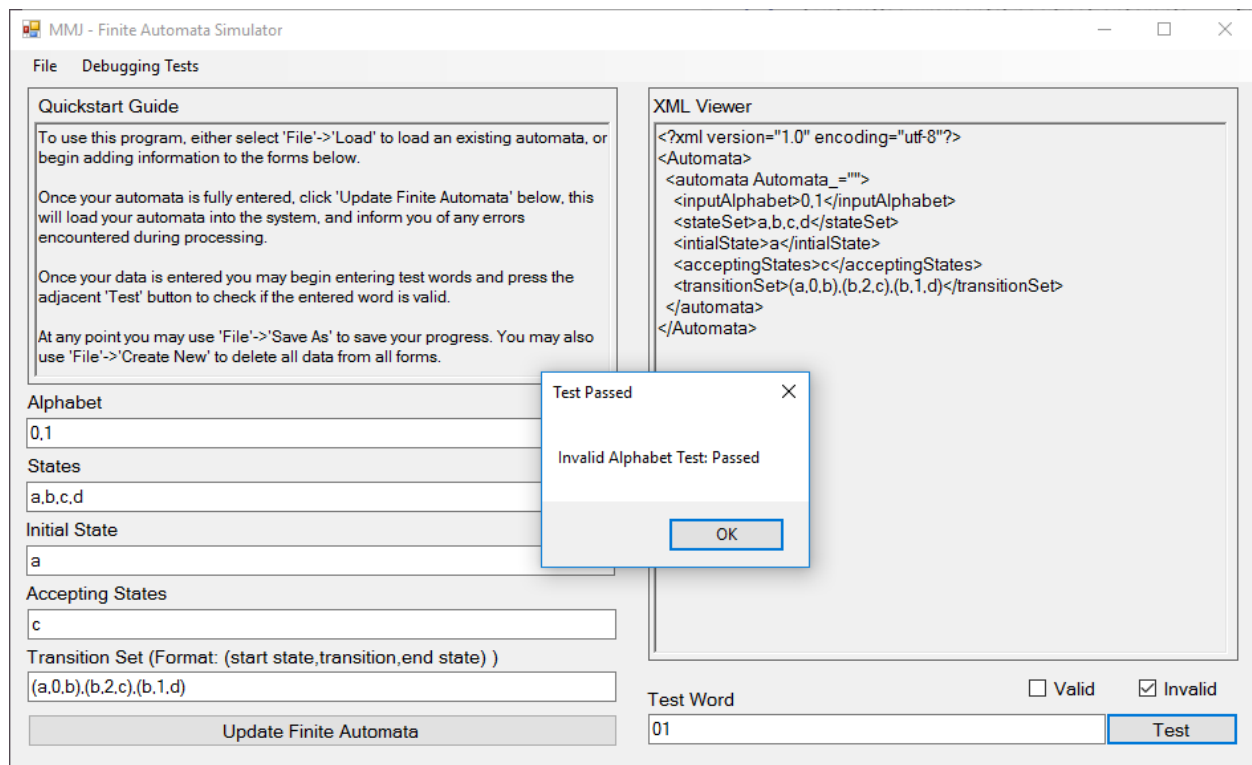
Standard Fail:



Expected Behavior: Invalid

This test will handle the case of passing a word that should not be accepted by an automaton object.

Invalid Alphabet:



Expected Behavior: Invalid

This test will handle the case of trying to go through a transition who's data field is not included in the alphabet (automaton is not represented right).