<p style="text-align:center"><u>Programming Assignment-6</u></p>

Akanksha Bansal   (akankshabansal90@gmail.com)
Gaurav Nanda   (gaurav324@gmail.com)
Prateek Agarwal   (prat0318@gmail.com)

FSMLite:

java FSMLite.Conform xxx

xxx is expected to be a xxx.state.violet file. Main() method calls the custom written State Parser , converts .violet to .violet.pl . And then adds state.conform.pl constraints at the end of the .violet.pl. Swipl is internally called to validate the constraints and spits out errors if present and exits.

java FSMLite.Convert xxx

It internally calls Conform to validate the .violet file. If correct, it uses 'Convert2Java.vm' to parse the .pl generated to generate .java files.

Important directories:
Inputs/ It contains the violet xml files for all the test cases of P5

Outputs/ It contains all the .java, .pl files created during script execution.

<u>Running Instructions:</u>
MDELite2.3/config.properties [currently set to MAC environment]

It contains SWIPL path location and has to adjusted as per the user's OS environment.

$ sh ./run.script

This script will compile the code for the above sections and then call functions to generate the java files from the .violet xml given in the argument. The output will indicate if the tests have run successfully or not.

Expected Output:
If any FSM violet XML doesn't follow the model constraints, it is expected to see a Conformance failure. Something like below is not a defect, but a correct expectation:

```
Conformance Failure
Exception in thread "main" java.lang.RuntimeException
      at CoreMDELite.GProlog.executeProlog(GProlog.java:115)
```