**Testing Outcomes**

The first test I made was to check the empty entry { } was accepted, which it was.

The next simplest test was whether the pair of empty strings was accepted **{" " : " "}** which again ran without error.

The next step I took was after defining a character as any Unicode character bar **“** and **/**. I checked whether these entries caused an error.

For example having the input.txt file set as: **{" " ": "abc"}** gives an error:

Syntax error for input symbol "QUOTE" spanning from null to null

Couldn't repair and continue parse for input symbol "QUOTE" spanning from null to null

Exception in thread "main" instead expected token classes are []

java.lang.Exception: Can't recover from previous error(s)

at java\_cup.runtime.lr\_parser.report\_fatal\_error(lr\_parser.java:392)

at java\_cup.runtime.lr\_parser.unrecovered\_syntax\_error(lr\_parser.java:539)

at java\_cup.runtime.lr\_parser.parse(lr\_parser.java:731)

at cup.example.Driver.main(Driver.java:9)

Similarly, the input of **{"\" : ""}** gives the more simple message of:

Illegal character: \

After testing a string of the type “a1” which included numbers and characters established that my original definition of a string did not allow for anything that had been previously defined to be included within it. Was therefore necessary to define a char as any of the things we defined so they can be used within a string.

The test of **"A": "test 89:12 " –** then gave no errors.

A more complicated example is given below which tested all possibilities including all the previously defined terminals within a string – which ran without issue.

{

"":{},

"true": true,

"null":null,

"false": "test 123 true, null false [] {}",

"A": "test 89:12 "

}

The next test I did was for the arrays:

{

"":{},

"true": true,

"null":null,

"false": "test 123 true , null false [] {} : ",

"A": "test 89:12 ",

"Array" : [1, 2,3, a, true, false, null, []],

“Array2" : []

}

This gave an error due to the character a not being contained by “ “-

Syntax error for input symbol "CHAR" spanning from null to null

Couldn't repair and continue parse for input symbol "CHAR" spanning from null to nullinstead expected token classes are [LSQ, LCURLY, NULL, NUMBER, TRUE, FALSE]

Exception in thread "main" java.lang.Exception: Can't recover from previous error(s)

at java\_cup.runtime.lr\_parser.report\_fatal\_error(lr\_parser.java:392)

at java\_cup.runtime.lr\_parser.unrecovered\_syntax\_error(lr\_parser.java:539)

at java\_cup.runtime.lr\_parser.parse(lr\_parser.java:731)

at cup.example.Driver.main(Driver.java:9)

The corrected version gave no errors.

{

"":{},

"true": true,

"null":null,

"false": "test 123 true , null false [] {} : ",

"A": "test 89:12 ",

"Array" : [1, 2,3, "a", true, false, null, []],

“Array2" : []

}

I also tested the various types of numbers, which displayed no errors:

{

"Value1":1,

"Value2":1.23,

"Value3":1e-2

"Value4":-1

}

The remaining tests I performed were to test whether the various special options that are accepted within a string that are preceded by a \. – which all run fine.

{

"\u": "\\",

"\"": "\b",

"\f": "\n",

"\/": "\r",

"\/\utf": ""

}

This means all combinations have been checked – An example of what is shown with the input.txt file after the driver class is run in eclipse with no errors is shown on the next page.

