David Rudolf
MSc Computer Science - COMPGC04

Testing

To be able to test the code, Java print functions were added with the associated non-terminal symbols in the parser so that whenever there is a value or a name/value pair in the input stream, the values and the name/value pairs will be printed out. The code was tested with some regular value and name/value inputs as well as the JSON example code that figured in the Coursework introduction. During the first tests it was found that the lexer did not support negative zero decimal values. Hence the necessity for the additional macro as mentioned in the system description.

There seems to be a problem when trying to use the forbidden '*u*' character. The JSON specification says that it should not be part of a string but whenever it is included within the lexer, the lexical analysis stops working. JFlex uses '*u'* as a Unicode escape sequence and therefore is cannot be part of the lexical rules. This also applies to '*x'* which is used as an ASCII escape sequence in JFLEX.

Input	Output	Reason

{
"coord": {
"lon": -0.13,
"lat": 51.51
},
"weather":[
{
"id":300,
"main" : "Drizzle",
"description" : "light intensity dizzle",
"icon" : "09d"
}],
"main" : {
"temp" : 280.32,
"pressure" : 1012,
"humidity": 81,
"temp_min" : 279.15,
"temp_max" : 281.15
}}
	name: "coord"
	name: "lon"
	value: -0.13
	name: "lat"
	value: 51.51
	value: null
	name: "weather"
	name: "id"
	value: 300.0
	name: "main"
	value: "Drizzle"
	name: "description"
	value: "light intensity dizzle"
	name: "icon"
	value: "09d"
	value: null
	value: null
	name: "main"
	name: "temp"
	value: 280.32
	name: "pressure"
	value: 1012.0
	name: "humidity"
	value: 81.0
	name: "temp_min"
	value: 279.15
	name: "temp_max"
	value: 281.15
		This JSON code was given as an example in the coursework description file therefore it was a good starting point for the testing.
		
		It encompasses both arrays and objects and different value types as well as nested objects and arrays.

{"david" : -123, "christmas" : "eve", "NuMber":0.0123, "NUMBER" : -0.123456 }	name: "david" value: -123.0 name: "christmas" value: "eve" name: "NuMber" value: 0.0123 name: "NUMBER" value: -0.123456	This tests the different types of letters and numbers that we may have in the input. Capital / non-capital letters. Integers, decimals, negative integers, negative decimals.
["#'@:*&^%$"]	value: "#'@:*&^%$"	This tests some Unicode characters as part of a string.
["#'@:*&^%$+=~?!"]	value: "#'@:*&^%$+=~?!"	This adds some more Unicode characters to the string.
{"is the earth flat?" : false, "is the earth round?" : true}	name: "is the earth flat?" value: false name: "is the earth round?" value: true	This tests if the lexer/parser recognises Boolean expressions as values.

{"What is the meaning of Life?" : null}	name: "what is the meaning of Life?" value: null	This tests the null value.