Assignment: Analytic Tableaux for Propositional Logic

The goal of this assignment will be to implement tableau method for determining the satisfiability of finite set of formulas belonging to propositional logic. In order to do so, you will be expected to implement the following

1. Provide an abstract class/signature for Logical formulas. For this assignment you should implement a class/structure for propositional formulas which should be a specific instance of your abstract class. The point of having the abstract class is to make your implementation extensible to other logics like first order logic. You should provide encode/decode function to convert XML input to formulas and vice versa. You should also implement a method for checking logical equivalence of formulas, which would be useful in removing redundant repetition of formulas in tableau sets. All these methods should be declared by the abstract class.

2. Provide an implementation for formula sets. You may use the logical equivalence checking to remove redundant formulas from your set. However you should not convert formulas to some normal form. (you may use normalization for equivalence checking however)

3. Implement a method *TableauRuleApplication()*, which takes a formula set as input, chooses a formula from the formula set based on the tableau rule application policy for the given logic, and generates new formula set(s) by applying the tableau rule on the chosen formula. Your method should also check to see if the input set is contradictory (i.e. it contains complementary literals in case of propositional logic) in which case it outputs $\{\bot\}$ and if the input set is not contradictory but can not be further extended it outputs $\{\top\}$.

4. Implement the procedure for generating tableau tree for a given formula set and determining if all the paths are closed (i.e. end with the leaf node $\{\bot\}$) or if there is some open path (branch ending with $\{\top\}$). This implementation can be done in generic way, as all the logic specific details can be handled with *TableauRuleApplication()* method for non-destructive tableau calculi.

Input-Output format

Your program is expected to take the XML input from a file *'input.xml'*, which would provide the set of formulas whose unsatisfiability needs to be determined. If unsatisfiable you are expected to output all the contradictory sets obtained

operator	XML tag	#child	Attributes
a	Letter	0	val="a"
$\vee$	Disjunction	2	
$\wedge$	Conjunction	2	
$\neg$	Negation	1	
$\rightarrow$	Implies	2	
$\leftrightarrow$	Equivalence	2	
sets	Sets	#sets	
set	Set	#elements	
set element	Element	1	

Table 1: XML format for input/output files

while closing paths. If satisfiable you are expected to output any one non-contradictory set obtained on the open paths which cannot be extended further. All output should be in XML form and must go to a file *'output.xml'*.

A sample input.xml and output.xml has been provided as an example. In the case of Implies tag, the formula given by first child *implies* the formula given by second child. Note that the input only has one set, so the root xml tag in input is Set, while the root xml tag in output is expected to be Sets.

You may choose any programming language of your choice for the implementation. Libraries for XML parsing are easily available for all major languages. However in order to facilitate automated testing you are expected to provide a bash script *'runAsgn1.sh'* which will run your program on the input.xml assumed to be present in the same directory as the script file and generate output.xml.

Note: Please keep in mind that you are expected to *extend* the implementation of this assignment to first order logic.