XML Validation using XML Schema

Name: Madhur Kapoor

USN: 1PI10CS139

Objective: Given an XML Schema as an Input, verify the given XML Document

Language: Python

Input: A list of node type objects occurring in the order in which they occur in the XML document.

Procedure:

* The input is taken in the form of a list of XML tags, occurring in the order of their occurrence in the XML Document.
* A blank sentinel node is created which serves as the head of the stack.(The initial top of stack).
* Iterate through the list of tags and create a selection structure based on the nesting depths of subsequent tags in the list.
* According to this selection,a deep copy of the nodes is placed appropriately in a tree structure under some encapsulating node and popped off from the stack i.e., the node list.
* If the nesting depth of a tag(node) is greater than the nesting depth of its predecessor,then

append it to the stack and make it the top of stack.

for example say, an instance of stack frame is (numbers represent the depth of the node)

1|2|3|4|5 (5 is the current top of stack)

once the closing tag for 5 is reached, the depth reduces to 4 again.

Thus,

1|2|3|4|5|4

Now, 4 is popped off the list and stack and 5 is placed as a child to 4 and is also popped off the stack afterwards. Making 4 as the new top of stack.

Similarly the tree is created.

* Once the closing tag for the current top of stack is encountered ,pop it off the stack and place the current top of stack under the node before that in the stack. (as in place 5 under 4 in this case and popped off).

* Once the XSD Tree is created, it is augmented , in the sense that it is degraded so as to bring it town to a level in which it can be compared with the original XML document tree. This operation reduces the the complexity of tree comparison.
* Search for the “element” tags in the XSD tree and place all the underlying element information in the element node as attributes (for syntax matching in the later phase)i.e. “complexType” and “sequence” is not included in the augmented tree structure as nodes instead placed as attributes in the parent element.
* The augmented tree only has element tags and all other information is present inside element tags as attributes.
* Also care has been taken for the order of these attributes and the order mismatch errors can be caught.