Application of Tree Automata in The Validation of XML Documents

Lajos Kollár

In recent years, XML has become a communication standard of the Web. Since XML is a metalanguage, it needs a schema to define concrete languages for particular applications. Different schema languages have appeared in the past few years, e.g., DTD, XML Schema, RELAX Core, TREX, and so on. In this paper we deal only with RELAX and TREX which are based on tree regular language theory.

All XML documents have a logical structure which forms a tree. Validation of an XML document is the process of checking whether the document corresponds to its schema or not. The validation of a document against its schema written in such a schema description language could simply be done by constructing a tree automaton which is to decide whether the tree representation of this document is generated by the given regular tree grammar or not.

The above mentioned schema languages have quite different characteristics from the viewpoint of Formal Language Theory. RELAX and TREX are appropriate for expressing any regular tree grammar, therefore they are more expressive than the others. Since the class regular tree languages is closed under union, intersection and difference, it has many advantages of using RELAX or TREX to describe the required schema. RELAX Next Generation (RELAX NG) combines the benefits of those two similar languages.

With the progress of time, new requirements could emerge against the application. Hence, schemas evolve to new schemas or new versions of old schemas. There could also be some kind of 'convergence' between applications of the same domain (e.g., on-line book stores): they could share information using the same schema. To achieve this goal, the integration of current schemas should be supported. That is exactly why closure of the communication language is so important.

Beyond these issues, the problem of 'dynamically changing schemas' is examined (which means that the grammar is changing while the application is running): we describe possible kinds of changes that can be applied to the schema. Based on Document Object Model (DOM), a Java implementation of an application which re-validates the document against the changed schema is also given.