

Commutative Grammars and Permutation Grammars

Martin Tomko
itomko@fit.vutbr.cz

Traditional grammars and automata work with strings in a strictly sequential manner. Among other things, this means that the order of symbols in a string is very important, and changing it will change the identity of the string itself.

In contrast with this approach, *commutative grammars* work on so-called *bags*, finite multisets of symbols which can be thought of as strings in which the order of symbols does not matter. These grammars can be shown to be equivalent to two established parallel processing models – Petri nets and vector addition systems.

Another model that goes against the traditional sequential processing of strings is *permutation grammars*, which are context-free grammars extended with so-called *permutation rules*, that is, rules such that their right-hand side is a permutation of the left-hand side. An example of a permutation rule would be $ABC \rightarrow BCA$. These grammars generate so-called *permutation languages*, the class of which turns out to be strictly between the classes of context-free and context-sensitive languages.

The purpose of this talk is to give a quick introduction to these two models, describe their basic properties and describe their relation to more traditional language models.