# Commutative and Permutation Grammars

David Holas, Tomáš Kocourek
Brno University of Technology (Czech Republic)
{xholas11,xkocou10}@stud.fit.vutbr.cz

**Abstract**

In the context of compilers, we usually apply context-free grammars ($CFGs$) as a formalism for parsing. Their more general version, context-sensitive grammars ($CSGs$), have greater generative power. However, this formalism is not as practical as $CFGs$, since there are no reasonably effective algorithms for parsing $CSGs$ in comparison to $CFGs$. To reach a compromise between $CFGs$ and $CSGs$, the Permutation grammars ($Perms$) are presented. They preserve the capabilities of $CFGs$, but they also allow us to extend this concept by *permutation rules* which significantly widen the generative power of $CFGs$.

In this lecture, we define a Parikh mapping and the notion of letter equivalence. Then, we can properly explain the permutation grammars and also take a closer look at the notion of base language. By doing this, it is pretty straight-forward to show the place of permutation grammars in the Chomsky hierarchy. Next, we discuss the possibility of imposing parametric restriction of length of the permutation rules, which allows us to build a hierarchy of $Perm_n$ grammars. At the end of the presentation, we state that $Perm_2 \subset Perm_3$ and $Perm_{4 \cdot n-2} \subset Perm_{4 \cdot n-1}$ which implies that $Perm_n$ grammars form an infinite hierarchy. The proof is left out because of its sheer complexity.

We also present the concept of commutative grammars ($CGs$) which are convertible to the formal model of Petri nets and vice versa. This type of grammar does not generate set of strings as $CFGs$ but set of *bags*, which correspond to the finite multisets of symbols. In this lecture, we define the idea of bags and operations over them. Then, we introduce commutative grammars, and we compare them to the classical grammars over strings.

**Keywords**

commutative grammar, permutation grammar, bag, Parikh image, permutation hierarchy