DESCRIPTION FOR THE GENERAL PUBLIC

Let's imagine, Dear Reader, that we are solving together an exceptionally exciting „Sudoku" puzzle. If we gave free rein to our imagination, we could further assume that each of us has his/her own preferred strategy for this kind of challenge. For example, I would solve the puzzle in a purely algorithmic way, by subsequently applying the following three rules: (1) *analyse the rows and fill in the cells, if possible*, (2) *analyse the columns and fill in the cells, if possible*, (3) *analyse smaller squares and fill in the cells, if possible*. Repeat, if the task is not finished. You would, however, immediately notice, that this "automatic" way of solving the task is highly inefficient, as even a cursory analysis of density of the rows, columns and squares might suggest a simpler—not as algorithmic, more distributed, but less laborious—procedure.

The aim of our project is to make use of this simple observation in the analysis of artificial systems performing more complex, but also more specialised tasks. Let us imagine an artificial system (*e.g.* a computer program) that solves a complicated problem (like "Sudoku"). The system analyses the structure of the problem, divides the problem into "subproblems" (such as, for instance, putting "4" in a selected row), and then for each of the subproblems it chooses means appropriate for its realization—in our "Sudoku case", on the basis of the density of numerals the system decides whether to apply rule (1), (2) or (3). The most important measure of "appropriateness of means" should be expressed in terms of the efficiency of the solution; availability of the selected methods could be also considered (*e.g.* if it is possible at all to fill in a given row at a given stage). In the last step, the system integrates the obtained results into the final solution of the initial problem (our "Sudoku" puzzle is solved).

The problem-solving schema that we have sketched above can be implemented in the realm of proving theorems of formal logic. The search for a proof is a special kind of problem-solving—it is a quest to decide whether a given theorem is true or false. The problem of proving theorems lies in the center of *proof theory*—a discipline traditionally conceived as a part of philosophical logic, pursued by means of formal logic and / or mathematics. Proof-theorists usually analyse the notion of proof by constructing formal deductive systems composed of rules that allow to "compute" the consecutive expressions of a formal language, and therefore to construct a proof. Traditional deductive systems are very constrained in their possibility of choosing "appropriate means". On the other hand, we know nowadays that various systems can solve the same problems with extremely diverse effectiveness. In other words, the efficiency of the methods used to solve a problem depends on the structure of the problem.

Distributive deductive system is a generalization of traditional deductive system which allows for a division of the initial problem into a series of subproblems. Each of the subproblems is further analysed by the means available in a deductive system best suited to the structure of the subproblem. As a result, the burden of the proof is distributed among different modules solving the subproblems.

The measurable result of the proposed research will be the implementation of distributive deductive systems for a collection of logical systems. The implementation will allow to analyse an important class of deductive reasoning. It will contain simple modules conducting proofs in selected deductive systems, in addition, it will be equipped with useful functionalities—such as optimisation of proof-search strategies. Furthermore, the use of computational methods in the analysis of traditional concepts of proof theory shall allow for a modern reflection on these concepts. The computational techniques are able to uncover hidden patterns, unavailable by the traditional means, thus changing our view on the very notion of proof.