Research project objectives/Hypothesis In his recent research the applicant successfully applied grammar-compression methods to problems in various fields of computer science. The developed recompression method switches the usual focus: instead of trying to exploit the underlying combinatorics of the problem, it analyses and changes the (implicit) representation of the problem, gradually simplifying it. In this project the applicant wants to push further the usage of such techniques: in fields in which it already proved to be useful as well as in related ones, in which such techniques were not used. Besides that, the applicant also wants to investigate problems related to succinct (compressed) representations of regular languages.

The most important result obtained using recompression is the decidability of *context unification*, which is a problem of solving equation (over terms) in which variables represent terms with a missing argument and the substitution for a variable uses the argument exactly once. Still, there are several variants and generalisations, which can be further investigated: the subproblems in which the number of variables is bounded still has unsettled complexity. On the other hand, if we allow regular constraints in the instance, the current proof fails. Lastly, so far the algorithm gives only answer to the decision problem, it does not provide a full description of all solutions.

Context unification is closely related to term rewriting and in fact the decidability of positive existential theory of one-step term rewriting (so a theory whose only allowed predicate says 't rewrites to s in one step') was shown by reduction to a fragment of context unification. Still, the decidability status of both more general fragments (existential and positive) remain unknown.

In word equations (in which we solve equations over strings and variables represent strings) this technique greatly simplified the existing proofs of decidability as well as sped up the algorithm for variant with only one variable. The running time of variant with two variables is $\mathcal{O}(n^5)$ and perhaps it can be lowered using the same technique. Also, the fragment in which only two variables are used (*quadratic*) is widely believed to be in NP (so is its counterpart in free groups), perhaps this can be shown using compression based techniques.

An intermediate step between free semigroup and free group is the free inverse monoid, in which some sort of inverse element exists. In general, solving equations in it is undecidable, however, the status of equations with only one variable remains unknown. Moreover, an equation in a free inverse monoid induces an equation in a free group, and if the former is satisfiable, so is the latter. The somehow converse question of extendibility (given a solution of the equation in the free group, does it have a corresponding solution in the free inverse monoid) is known to be in $\mathsf{DEXPTIME}$, but no lower bound for the problem is known.

Most of algorithmic research for compressed data was carried out for strings. While the grammar compression was generalised to trees, only a small portion of algorithms was generalized from strings to their trees counterparts, usually due to increased combinatorial difficulties. The applicant hopes that the recompression based approach, which ignores such combinatorics, will allow several such algorithms: data structure for testing equality of dynamic trees, algorithm for testing equivalence of non-linear tree grammars and submatching (so looking for left-hand sides of a string rewriting rule) in compressed trees.

Another popular model of compression are Huffman codes. Their decoding works badly in presence of errors, as one error may corrupt the whole decoding procedure. One of proposed solutions is adding synchronising words, which ensure that further decoding is correct, regardless of previous errors. The applicant plans to improve known upper bounds on the length of shortest synchronising words and also to establish the computational complexity of calculating the length of the shortest such words.

Lastly, the applicant wants to consider modern approaches to decomposition of languages: in order to speed up the processing of a formal language L, say during automata-based model checking, one can decompose this language according to some operation $*$, so as $L = L_1 * L_2 * \ldots * L_k$, process each part in parallel and combine the results. The applicant wants to develop a decision procedure for answering, whether a regular language can be decomposed according to intersection. On the other hand, the descriptional complexity community investigates the same problem from a different angle: given the size of automata recognising $L_1, L_2, \ldots, L_k$, what are the possible sizes of the combined language L? The answers to this question are not known even for common operations (concatenation, reversal, Kleene star) for small alphabets.

Research methodology A traditional mathematical-proof methodology applied to computer science is used. The algorithms' performance (so running time, space usage) is evaluated in the worst-case scenario. The claimed properties of algorithms are shown using mathematical proofs. Possible experiments are used only to provide heuristical data for developing proofs.

Research project impact The project aims at solving several important open questions from various areas of computer science. Thus a success implies an important development in respective fields. As this is a theoretical work, the progress is mostly in understanding, the applicant cannot point any immediate applications of the results.