The main aim of my report is to draw attention to some ideas which were successfully exploited for constructing fast algorithms for string-matching and related problems, and to discuss ways of making use of these ideas in more complicated identification problems. The ideas to be discussed are the idea of identifier and the idea of substructure automorphism. They may prove to lead to essential decreasing of upper bounds on the computational complexity of the problems. Lower bounds remain outside the discussion. The recent progress in investigating lower bounds on the complexity of concrete problems is at best implicit. My exposition pursues purely theoretical objectives and the choice of problems for discussion is entailed by theoretical considerations and not by applications.

By the complexity I mean the (worst-case) time complexity for address machines introduced by Slisenko [1]-[2] (in Angluin and Valiant [3] a similar model is named RAC), i.e. random access machines with relatively bounded registers – the length of computer words is not greater than the binary logarithm of the time (plus some function of lower order, if necessary). More or less detailed description of address machines can be found in Slisenko [2]. The choice of computational model is essential for "low-level" complexity. If we wish to stay on the firm ground of practicalness then this area of computational complexity becomes one of the most important. And the address machine model seems to be the most adequate model (among those used in computational complexity) for computations by one processor in homogeneous random access memory. The place of address machines among such models as Turing machines and Kolmogorov algorithms (or their slight generalization named "storage modification machines" by
Schönhage [4] is as follows. Let $AM(\varphi)$ be the class of functions (or predicates) computable by address machines with the time complexity not greater than $\varphi$, $TM_n(\varphi)$ - the similar class for Turing machines with $n$-dimensional tapes (one can take either the class of one-head machines or of multi-head ones), $KA(\varphi)$ - the similar class for Kolmogorov algorithms or storage modification machines. Within these notations: $TM_n(\varphi) \subseteq KA(C, \varphi) \subseteq AM(C, \varphi) \subseteq TM_1(C, \varphi^2 \log^3 \varphi)$;
$TM_n(\varphi) \subseteq AM(C, \varphi (\log \varphi)^{-1/\alpha})$, where $C_i$'s are constants (the first inclusion is due to Schönhage [4], the last is virtually some generalization of a theorem from Hopcroft, Paul and Valiant [5]).

As a starting point for the discussion I take the following "embedding framework" for representation of the problems to be considered. This framework is wider than I really need but it can play a certain organizing role. Let $G_i = (N_i, S_i), i = 1, 2,$ be two "graph-like" structures, where $N_i$ is a finite set of nodes, and $S_i$ is a finite set of links (i.e. partial mappings $S_i \rightarrow S_i$) and functions (i.e. partial mappings $S_i \rightarrow$ some set of words, e.g. integers). "Straight" embedding problem looks as follows. There are given mappings (or embeddings) $f_k: N_i \rightarrow N_j, 1 \leq k \leq m$, which are injective as a rule. And we have some "global" criterion $\Psi$ of the quality of a set of embeddings. The problem consists in computing $\Psi(G_1, G_2, \{f_k\}_m)$. Surely, the particular $\Psi$'s below will be rather easy to compute. Several "inverse" problems correspond to a given "straight" problem, e.g. for a given $G_i$ and a value of $\Psi$ (or some its property, such as "to be maximal") we are to find $G_2$ and $f_k$ of a certain type.

One of the simplest embedding problems, namely, string-matching, will illustrate the main ideas discussed in this report. We consider strings in some alphabet $A = \{a_1, ..., a_p\}$ and are interested in recognizing the set
\[
\{u \ast v; v \text{ is a substring of } u\}
\] (1)

In an input $u \ast v, \ast \notin A$, we discern the text $u$ and the pattern $v$. Now we represent the problem within the embedding framework. The length of $w$ is denoted by $|w|$ and the $i$-th letter (character) in $w$ by $w(i)$. Let $N_1 = \{1, 2, ..., |u|\}$, $N_2 = \{1, 2, ..., |v|\}$ and $S_1, S_2$ describe the string structure on $N_1$ and $N_2$ (in fact, $S_1$ and $S_2$ will be incorporated in $f_k$), and $f_k(j) = k + j - 1$. Then

$$\Psi(u, v, \{f_k\}) = \max_k \{\phi(u, v, f_k)\},$$
where
$$\phi(u, v, f_k) = \max_i \{i \leq |v|; \exists j \leq i: v(j) = u(f_k(j))\}.$$