Notation. By $L = L[L_{adv}](A)$ we denote the fact, that A decides the language L with the advisory information, that the input belongs to L_{adv}. Another way of looking at this fact is, that $L[L_{adv}](A) = L(A) \cap L_{adv}$.

Example 1. Let $L = (a^6)^*$ and $V = (Q, \Sigma, \delta, q_0, F)$, where $Q = q_0, q_1, q_2$, $\Sigma = \{a\}$, $F = \{q_0\}$ and $\delta(q_i, a) = q_{i+1 mod 3}$ for $i = 0, 1, 2$. Moreover, let $L_{adv} = (a^2)^*$. Then, althought $L \neq L(V)$, it is easy to see, that $L = L[L_{adv}](V)$.

Definition. Let T be an a-transducer and L a language. Then $T^{-1}(L)$ is the set of all words, such that it images belong to L. Formally

$$T^{-1}(L) = \{w | T(w) \subseteq L\}.$$

Description of the framework: we have a decider D, which should construct a (deterministic?) finite automaton for the language L_{dec}. Moreover, we have an advisor (oracle) O. Now, O sends D a dual information: an a-transducer T and a regular language L_{adv}. This information forms a promise, that if D transforms a correct input (that is, input from L_{dec}) using T, it will belong to L_{adv}. Now, D has two possibilities:

1. it trashes the information from O and constructs an automaton A for L_{dec} "from scratch"
2. it creates a simpler automaton A', such that $L[T^{-1}(L_{adv})](A') = L_{dec}$.

Moreover, L_{adv} has to be verifiable by a finite automaton V.

Definition. The *state complexity* of an a-transducer $T = (Q, \Sigma_1, \Sigma_2, H, q_0, F)$ (a finite automaton $A = (Q, \Sigma, \delta, q_0, F)$), denoted by $\mathscr{C}_{state}(T)$ ($\mathscr{C}_{state}(A)$), is the number of its states. Formally

$$\mathscr{C}_{state}(T) = |Q|.$$

Definition. The *state complexity* of a regular language L, denoted by $\mathscr{C}_{state}(L)$, is the state complexity of its minimal (deterministic?) finite automaton. Formally

$$\mathscr{C}_{state}(L) = min\{\mathscr{C}_{state}(A) | L(A) = L\}.$$

If L is not regular, then $\mathscr{C}_{state}(L) = \infty$.

Definition. A language L_{adv} with an a-transducer T is an *effective advice with regard to* L_{dec}, if there exists an automaton A', such that $L_{dec} = L[T^{-1}(L_{adv})](A')$ and $\mathscr{C}_{state}(A') + \mathscr{C}_{state}(T) + \mathscr{C}_{state}(L_{adv}) \leq \mathscr{C}_{state}(L)$.

Example 2. Let $L_{dec} = \{a^{12k} | k \geq 0\}$, T be an one state a-transducer computing the identity and $L_{adv} = \{a^{2k} | k \geq 0\}$. D can now construct a simpler finite automaton A' for the language $L_{simple} = \{a^{6k} | k \geq 0\}$. Clearly, $\mathscr{C}_{state}(A') + \mathscr{C}_{state}(T) + \mathscr{C}_{state}(L_{adv}) = 6 + 1 + 2 \leq 12 = \mathscr{C}_{state}(L_{dec})$, which means, that L_{adv} with T is an effective advice with regard to L_{dec}.

Example 3. Let $L_{dec} = \{a^{12k} | k \geq 0\}$. Let $T = (\{q_0, q_1\}, \{a\}, \{a\}, H, q_0, \{q_0\})$, where $H = \{(q_0, a, a, q_1), (q_1, a, \epsilon, q_0)\}$ and $L_{adv} = \{a^{2k} | k \geq 0\}$. It is easy to see, that $T^{-1}(L_{adv}) = \{a^{4k} | k \geq 0\}$. D can now construct a simpler finite automaton A' for the language $L_{simple} = \{a^{3k} | k \geq 0\}$. Clearly, $\mathscr{C}_{state}(A') + \mathscr{C}_{state}(T) + \mathscr{C}_{state}(L_{adv}) = 3 + 2 + 2 \leq 12 = \mathscr{C}_{state}(L_{dec})$, which means, that L_{adv} with T is an effective advice with regard to L_{dec}.

Two interesting questions arise. The first is, for given language L and a-transducer T, how to get the language $T^{-1}(L)$? The answer was quite easy to find in previous two examples (and, in fact, for all languages in form $\{(a^k)^+\}$ and a-transducers, which just manipulates the number of symbols a).

Another question is in some sense the inverse perspective of this problem. We have a fixed language L and we want to transform it to a language L_{adv}. Since we want to minimize the complexity of the advice, our question is, what is the minimal state complexity of an a-transducer T, such that $T(L) = L_{adv}$. We address this question in Section [number of complexity section].