Definition. A sequence of generalized sequential machines $\mathcal{G} = \{G_n : n \in \mathbb{N}\}$ is called *logspace uniform*, if there is a deterministic logspace-bounded Turing machine, which for all n, outputs the description of G_n on the input 1^n.

Definition. A *translator* is a logspace(?) uniform sequence of generalized sequential machines.

Description of the framework: Let us have a sequence of gsm $\mathcal{G} = \{G_n : n \in \mathbb{N}\}$ and a deterministic finite automaton(?) V (verifier), which has to decide, wheter an input w belongs to a language L_{dec}. There is an advisor A, which has some information of the input w (the information is in the form of a language L_{inf}). L_{inf} is sent to V together with an index i to the sequence $\mathcal{G}$. V then transforms L_{inf} using a gsm G_i and gets the language L_{adv}. Now, V knows, that w belongs to L_{adv} and this can possibly help to reduce its complexity.

Definition. The *state complexity* of a gsm $G = (Q, \Sigma, \Delta, \delta, \lambda, q_1)$, denoted by $\mathscr{C}_{state}(G)$, is the number of its states. Formally

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

Notation. By $L = L[L_{adv}](V)$ we denote the fact, that V decides the language L with the advisory information, that the input belongs to 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. For a fixed sequence $\mathcal{G}$, a language L_{inf} with an index i is an *effective advice with regard to* L_{dec}, if there exists a verifier V with k states, such that $L = L[L_{adv}](V)$, the minimal DFA for L_{dec} has l states and $\mathscr{C}_{state}(G_i) < l - k$.

Example 2. Let $\mathcal{G}$ be a sequence of gsm, where G_i is a gsm, that computes a bitwise XOR of the input and a key k, which is obtained as follows: take a binary representation of i and remove the leading 1 (otherwise it would not be possible to have keys with initial sequence of zeros). If the input is longer than the key, we compute the bitwise XOR with k^*.

Lemma 1. If a key k of length n is nonperiodical (it has not a form k'^i for $i > 1$), then a gsm from example 1 with key k has at least n states.

Proof. Let $G = (Q, \Sigma, \Delta, \delta, \lambda, q_1)$ be a gsm, where $Q = \{q_1, ..., q_k\}, \Sigma = \Delta = \{0, 1\}, \delta(q_i, a) = q_{(imodk)+1}$ and $\lambda(q_i, a) = k_i \oplus a$ for all $a \in \Sigma$ and $1 \leq i \leq n$ (where k_i is the i-th bit of the key k). It is easy to see, that G computes the correct output.

Now, we would like to show, that G is the minimal gsm for this sequential function. Let G' be a gsm with $n - 1$ states, such that $G'(w) = G(w)$ for all $w \in \{0, 1\}^*$. Let us take a look at a computation of G' on an input 0^n - so the output should be k. G' clearly uses at least one of its states to output two (or more) letters (not necessarily consecutive).

TODO: finish

□

Lemma 2. For each n and l, which is a divisor of n, there exists a gsm G_i from Example 1, such that G_i uses a key $k, |k| = n$ and $\mathscr{C}_{state}(G_i) = l$.

Proof.

TODO: periodic keys

□

Corollary 2.1. Let $f_{\mathcal{G}}$ be a function defined as follows: $f_{\mathcal{G}}(i) = \mathscr{C}_{state}(G_i)$. Then, $f_{\mathcal{G}_\rangle}$ is not monotonic.

Example 3. Let $\mathcal{H}$ be a sequence of gsm, where H_k transforms the input from a k-ary alphabet $a_1, a_2, ..., a_k$ to a binary alphabet $0, 1$, using a standard encoding (i. e. a_i is encoded as binary encoded number i with added leading zeros, such that the length of the code is $\lceil log_2 k \rceil$).

Is is easy to see, that such gsm needs only one state, which transforms letters a_k to its binary code.

For this reason, it is sometimes convenient to measure the complexity not only by state count, but rather by complexity of the output function. We will call this measure the output complexity.

Definition. The *output complexity* of a gsm $G = (Q, \Sigma, \Delta, \delta, \lambda, q_1)$, denoted by $\mathscr{C}_{out}(G)$, is the sum of output length of its transitions. More formally

$$\mathscr{C}_{out}(G) = \sum_{q \times a \to w \in \lambda} (1 + |w|)$$

Lemma 3. For $1 \leq i \leq k$ and H_i from Example 2, $\mathscr{C}_{out}(H_i) = i * \lceil \log i \rceil$.

Proof.

TODO: add proof

□.

Corollary 3.1. Let $f_{\mathcal{H}}$ be a function defined as follows: $f_{\mathcal{H}}(i) = \mathscr{C}_{out}(H_i)$. Then, $f_{\mathcal{H}}$ is monotonic.