Administrativia: intermittent discussions with Arthur Rabinovich, Saminul Haque, Vincent Huang, Ming Feng Wan, extra material consulted:
`https://concurrency.cs.uni-kl.de/documents/ConcurrencyTheory_SS_2015/languages_week_10_11_12_13_14.pdf`

Pset IX

For all $k \in \mathbb{Z}^+$, let $\Sigma_k = \{1, \dots, k\}$ and let $L_k = \{x \in \Sigma_k^* \mid \exists a \in \Sigma_k.(a \text{ does not occur in } x)\}$.

Problem.

For all $k \in \mathbb{Z}^+$, construct a nondeterministic finite automaton N_k with $k+1$ states such that $\mathfrak{L}(N_k) = L_k$.

Solution.

Let $k \in \mathbb{Z}^+$ be arbitrary, and denote the set $[1, k] \cap \mathbb{Z}^+$ as I.

Let $Q = \{q_i \mid i \in I \cup \{0\}\}$ be a set of k states, let q_0 be the initial state and and let $\Sigma = \Sigma_k$.

Consider a string "123…k" and its cycle shifts with the last symbol truncated. Suppose we do not care about the ordering of the first three digits and want to count how many unique combinations there are. Since this is a problem of choosing $k-1$ elements out of k elements, we know that $\begin{pmatrix} k \\ k-1 \end{pmatrix} = k$ is the number of possible combinations. Since $|Q| = k+1$, let $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ be such that there are $k-1$ arrows from q_0 to each q_j for $j \in I$ in such a way that this transitions are labelled with numbers in $I - \{j\}$.

Take $\{q_i \mid i \in I\}$ as the set of accept states.

In this way, we have defined a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ which has $k+1$ states.

Note that this NFA represents N_k, since any string which includes all elements from Σ_k is rejected: there is no path which transits through the edges labelled with all the integers in I, so that q_0 plays a role of a *sorting* state, after which the entire set of allowed strings in one of k *branches* is well-defined.

By generalisation, $\mathfrak{L}(N_k) = L_k$. □

Problem. Prove that every deterministic finite automaton D_k such that $\mathfrak{L}(D_k) = L_k$ has at least 2^k states.

Solution.

For any $n \in \mathbb{Z}^+$, let $P(n) =$ "Any finite state automaton D_n such that $\mathfrak{L}(D_n) = L_n$ has at least 2^n states".

Note that $P(1)$ holds, since if $\Sigma_1 = \{1\}$ and there is only one state in the DFA corresponding to L_1, then there it is an initial and accepting note at the same time, while in nonempty DFAs there is always a transition labelled with an element from the chosen alphabet. Thus, there must be at least two states, as required.

Suppose now the claim holds for all positive integers less than some $k \in \mathbb{Z}^+$. We prove that $P(k)$ holds, and thus there are at least 2^k states in D_k.

Since we have already shown that $P(1)$ holds, without loss of generality assume that $k \geq 2$.

Since D_k is a DFA with the maximum length of the accepted string equal to $k-1$, there exists a state q_t such that there are k transitions each leading to a candidate for an accepted string of length $k-1$. Call these paths *subbranches*. Since each such a subbrunch accepts a string of length less than or equal to $k-1$, we know by inductive hypothesis that each of them requires at least 2^{k-1} states. Therefore, there are at least $k \cdot 2^{k-1} \geq 2 \cdot 2^{k-1} = 2^k$ states required in total, which is exactly the claim in case $n = k$.

Therefore, for all $n \in \mathbb{Z}^+$ we have $P(n)$ by strong induction.

□

Problem.

Prove that the family of regular languages is closed under finite state transductions.

Solution.

We prove that, for all regular languages $R \subseteq \Sigma^*$ and all functions $T : \Sigma^* \to \Gamma^*$ defined by finite state transducers $(Q, \Sigma, \Gamma, \delta, \tau, q_0)$, $T(R) = \{T(x) \,|\, x \in R\} \subseteq \Gamma^*$ is regular.

To prove it, we construct a nondeterministic finite automaton N with λ transitions, which, given a string $y \in \Gamma^*$ as input, guesses a string $x \in \Sigma^*$ one letter at a time, checks that both $T(x) = y$ and $x \in R$, and, if so, accepts y. The output T generates as a result of reading one letter is put into a buffer that is part of the state. In this way, we construct N which accepts $T(R)$.

First, let R be an arbitrary regular language which is a subset of Σ^*, suppose that D is a corresponding DFA and let M be an arbitrary transducer from Σ^* to Γ^* in the form $(Q, \Sigma, \Gamma, \delta, \sigma, q_0)$, with T an arbitrary transduction in M.

Let q_0, the initial state of D be equal to q_0', the initial state of N.

We prove first that there exists an alphabet Z, homomorphisms $\pi : Z^* \to \Sigma^*$ and $\tau : Z^* \to \Sigma^*$ and a regular language $K \subseteq Z^*$ such that $T = \{(\pi(y), \tau(y)) \,|\, y \in K\}$.

Let Z be a set of all pairs which serve as labels of each transduction of T.

In this way, M can be interpreted as a DFA D' over Z. Let K be a language corresponding to D'.

For all $(x, y) \in Z$, define $\pi : Z^* \to \Sigma^*$ and $\tau : Z^* \to \Sigma^*$ as $\pi((x, y)) = x$ and $\tau((x, y)) = y$.

In this way, by construction, we obtain that any pair (m, n) is in T if and only if there exists a sequence of states with transductions $(x_1, y_1), \ldots, (x_n, y_n)$ in M such that $m = x_1 \cdots x_n$ and $n = y_1 \cdots y_n$, which is equivalent to the sequence of states in D', since $m = \pi((x_1, y_1) \cdots (x_n, y_n))$ and $n = \tau((x_1, y_1) \cdots (x_n, y_n))$. Thus, there exists some $t \in K$ with $\alpha(t) = m$ and $\beta(t) = n$.

Moreover, if $T = \{(\pi(y), \tau(y)) \,|\, y \in K\}$, every edge in a finite-state automaton D' accepting K can be turned into an edge in the corresponding finite-state transducer.

This means that transducers can be translated into the language of DFA.

Since we have already proved that homomorphisms, inverse homomorphisms and intersections of regular languages preserve the regularity, we know that the family of regular languages is closed under finite state transductions. □