Languages, automata and computations II
Solutions to star exercises I

Konrad Majewski

2. Characterization of ω-regular languages

Let us denote the condition "if $u_i \sim v_i$, then $u_1u_2\ldots \in L \iff v_1v_2\ldots \in L$" as *substitution property*, and the condition "if $u_1 \sim v_1$ and $u_2 \sim v_2$, then $u_1u_2 \sim v_1v_2$" as *concatenation property*.

Now, we may proceed to the proof.

$\implies$ Let L be an ω-regular language. We need to show that there exists an L-compatible relation. Let $\mathcal{A}$ be a Büchi automaton recognizing language L and let Q, Δ be the sets of states and transitions of $\mathcal{A}$, respectively.

We define a function $f : \Sigma^* \to 2^{Q\times Q\times 2^\Delta}$ such that $(q, q', \Delta') \in f(w)$ if and only if there exists a run of automaton $\mathcal{A}$ on word w which starts in state q, ends in state q' and uses exactly transitions from Δ'.[1] Let us denote such a situation by $q \xrightarrow{w,\ \Delta'} q'$. Function f corresponds to an equivalence relation: $u \sim v \iff f(u) = f(v)$.

Clearly, this relation has finite index since codomain of f is finite. It remains to show that relation $\sim$ is L-compatible.

Substitution property. Let us consider infinite sequences (u_i), (v_i) of finite words such that $u_i \sim v_i$. Assume that $u = u_1u_2\ldots \in L$. This means that there exists a run of automaton $\mathcal{A}$ over word u such that infinitely many words u_i contain an accepting transition on their run. Since $u_i \sim v_i$ we can find a run over word $v = v_1v_2\ldots$ such that each word v_i starts and ends in the same states as the word u_i and words v_i are using exactly the same transitions of $\mathcal{A}$ as words u_i. This run certifies that $v \in L$, as desired.

Concatenation property. Let us consider words $u_1, u_2, v_1, v_2 \in \Sigma^*$ such that $u_1 \sim v_1$ and $u_2 \sim v_2$. Assume that there is a run of automaton $\mathcal{A}$ over word u_1u_2 of the form $q \xrightarrow{u_1u_2,\ \Delta'} q'$. Let q'' be a state on this run such that $q \xrightarrow{u_1,\ \Delta_1} q''$, $q'' \xrightarrow{u_2,\ \Delta_2} q'$ and $\Delta_1 \cup \Delta_2 = \Delta'$. Since $u_1 \sim v_1$ and $u_2 \sim v_2$ we can find runs of automaton $\mathcal{A}$ over words v_1, v_2 of the form $q \xrightarrow{v_1,\ \Delta_1} q''$ and $q'' \xrightarrow{v_2,\ \Delta_2} q'$. Concatenating these runs we obtain a run $q \xrightarrow{v_1v_2,\ \Delta'} q'$. Hence, $f(u_1u_2) \subseteq f(v_1v_2)$. Similarly, $f(v_1v_2) \subseteq f(u_1u_2)$, and thus $f(u_1u_2) = f(v_1v_2)$ which means that $u_1u_2 \sim v_1v_2$.

$\impliedby$ Let $\sim$ be an L-compatible relation with finite index. Let us denote by $r_1, r_2, \ldots, r_m \in \Sigma^*$ representatives of all classes of equivalence of relation $\sim$. We need to show that language L is ω-regular. First, we will prove two lemmas:

[1] State q does not need to be initial.

Lemma 1. Let $w = a_1a_2\ldots$ be an ω-word where $a_i \in \Sigma$. Then there exists its partition $w = vu_1u_2\ldots$ into finite words such that $u_i \sim u_j$ for each $i, j \in \mathbb{N}$.

Proof. We define a complete graph G on the set $V(G) = \mathbb{N}$. We color the edges of graph G by m colors as follows: for edge (i, j) (where $i < j$) we choose color k if and only if it holds $a_ia_{i+1}\ldots a_{j-1} \sim r_k$. By infinite Ramsey theorem we obtain that there exists a monochromatic clique (of some color l) in graph G. Let $t_1 < t_2 < t_3 < \ldots$ be an infinite sequence of numbers corresponding to the vertices of such a clique. Then words $v = a_1a_2\ldots a_{t_1-1}$ and $u_i = a_{t_i}a_{t_i+1}\ldots a_{t_{i+1}-1}$ for $i \in \mathbb{N}$ form the desired partition of word w because we have $u_i \sim r_l$ for all i, and thus $u_i \sim u_j$ for all i, j. □

Lemma 2. Let $r_k \in \Sigma^*$ be a representative of an equivalence class of relation $\sim$. Then, language $L_k = \{w \in \Sigma^* \mid w \sim r_k\}$ is regular.

Proof. We define a DFA recognizing language L_k as follows: we put one state corresponding to each equivalence class $[r_i]_\sim$. The initial state is $[\varepsilon]_\sim$ and the accepting state is $[r_k]_\sim$. All transitions are of the form $[r_i] \xrightarrow{a} [r_ia]$ for $a \in \Sigma$. Let us observe that such an automaton keeps track of an equivalence class of a word that was read so far. Indeed, the transitions preserve this invariant because if we read word $u \sim r_i$, then after reading the next letter $a \in \Sigma$ we move to state $[r_ia]$, and by concatenation property of $\sim$ we have $ua \sim r_ia$. □

Having proved the lemmas, we are ready to construct a Büchi automaton $\mathcal{A}$ recognizing language L. Let $w \in \Sigma^\omega$ be an input ω-word and let $w = vu_1u_2\ldots$ be its partition from Lemma 1. Let us say that $v \sim r_s$ and $u_i \sim r_t$. Then, by the substitution property of relation $\sim$ we have $w \in L$ if and only if $r_sr_t^\omega \in L$.

The automaton $\mathcal{A}$ starts with guessing classes $[r_s]$ and $[r_t]$ but it considers only these for which $r_sr_t^\omega \in L$ – this is realized by having at most m^2 copies of different automata. Let us say that it chose classes $[r_{s'}]$ and $[r_{t'}]$. The first part of automaton $\mathcal{A}$ recognizes regular language $\{v' \in \Sigma^* \mid v' \sim r_{s'}\}$ as in the Lemma 2. However, instead of having an accepting state we can move from this state by ε-transition to the second part of our automaton. We see that the first part of automaton $\mathcal{A}$ corresponds to the choice of prefix v of our word w such that $v \sim r_{s'}$. The second part of our automaton recognizes regular language $\{u' \in \Sigma^* \mid u' \sim r_{t'}\}$. Again, instead of having an accepting state q_a we add an accepting ε-transition from state q_a to the *initial state* of the second part of automaton $\mathcal{A}$. All ε-transitions that occurs here can be eliminated as in the case of finite automata. [2]

Summing up, we see that the language of such an automaton $\mathcal{A}$ is:

$$L(\mathcal{A}) = \{vu_1u_2\ldots \mid v \sim r_{s'},\ u_i \sim r_{t'},\ r_{s'}r_{t'}^\omega \in L\}$$

By the substitution property of relation $\sim$ we have $L(\mathcal{A}) \subseteq L$ and by the Lemma 1. we have $L \subseteq L(\mathcal{A})$, and thus $L(\mathcal{A}) = L$. □

[2] If we eliminate an accepting ε-transition then we mark added *skip* edges as accepting.

3. Fixed ambiguous automata

We will show that there exists the desired algorithm. This will be a consequence of the following lemma:

Lemma. Let $\mathcal{A}$ be a k-ambiguous automaton. Then, we can construct in polynomial time an automaton $\mathcal{A}'$ which is unambiguous, and which satisfies $L(\mathcal{A}') = L(\mathcal{A})$.

Let us observe that to decide universality of k-ambiguous automaton $\mathcal{A}$ it is enough to construct automaton $\mathcal{A}'$ from the lemma and run a polynomial algorithm for universality of unambiguous automaton $\mathcal{A}'$.

Proof of lemma. Let Q be a set of states of automaton $\mathcal{A}$ and assume that we fix some linear order on it. Our automaton $\mathcal{A}'$ simulates k copies of automaton $\mathcal{A}$ and the run of $\mathcal{A}'$ is accepting if there are k different runs of $\mathcal{A}$ sorted lexicographically. Clearly, if automaton $\mathcal{A}$ is k-ambiguous, then automaton $\mathcal{A}'$ is unambiguous.

Now, we describe details of automaton $\mathcal{A}'$. Its set of states is $(Q^k \times \{0,1\}^{k-1}) \cup \bot$. [3] The coordinates Q^k keep track of k runs of automaton $\mathcal{A}$ and coordinates $\{0,1\}^{k-1}$ indicate whether there was a difference between consecutive runs (i-th such coordinate is 1 whenever runs i and $i+1$ differs). Additionally, we move to state $\bot$ whenever the first difference between runs i and $i+1$ breaks lexicographical order. It is easy to see that we can construct transitions of automaton $\mathcal{A}'$ which preserve all mentioned above invariants. The initial states of $\mathcal{A}'$ are all states of the form $(i_1, \ldots, i_k, \varepsilon_1, \ldots, \varepsilon_{k-1})$ where $i_1 \leqslant i_2 \leqslant \ldots \leqslant i_k$ are initial states of automaton $\mathcal{A}$ and $\varepsilon_j = 1$ if $i_j \neq i_{j+1}$. Finally, the accepting states of $\mathcal{A}'$ are states of the form $(f_1, \ldots, f_k, 1, \ldots, 1)$ where f_j are accepting states of automaton $\mathcal{A}$. □

4. Co-finiteness of UFA

We will prove the following lemma:

Lemma. Let $\mathcal{A}$ be an unambiguous automaton with n states and assume that language $\Sigma^* \setminus L(\mathcal{A})$ is finite. Let w be the longest word such that $w \notin L(\mathcal{A})$. Then $|w| \leqslant n$.

Before proceeding to the proof we describe how above lemma relates to our problem. Given unambiguous automaton $\mathcal{A}$ with n states we want to construct in polynomial time unambiguous automaton $\mathcal{A}'$ with the following property:

$$L(\mathcal{A}') = L(\mathcal{A}) \cup \{u \in \Sigma^* : |u| \leqslant n\}$$

Then, by the Lemma we conclude that language $\Sigma^* \setminus L(\mathcal{A})$ is finite if and only if $L(\mathcal{A}') = \Sigma^*$. Moreover, we know that the latter condition can be decided in polynomial time since automaton $\mathcal{A}'$ is unambiguous.

Let Q be a set of states of automaton $\mathcal{A}$. We build an automaton $\mathcal{A}'$ with states $(Q \cup q_{\text{short}}) \times \{0, 1, \ldots, n, \infty\}$. The transitions of automaton $\mathcal{A}'$ are of the following form:

- $(q_1, i) \xrightarrow{a} (q_2, i+1)$ whenever there is a transition $q_1 \xrightarrow{a} q_2$ in automaton $\mathcal{A}$.
- $(q_{\text{short}}, i) \xrightarrow{\Sigma} (q_{\text{short}}, i+1)$

[3] Since we consider k as fixed the number of such states is of polynomial size with respect to the number of states from Q.

(in both cases if the second coordinate exceeds n we put ∞ on this coordinate)

We define initial states of automaton $\mathcal{A}'$ as:

$$\{(q_i, 0) \mid q_i \text{ is an initial state of } \mathcal{A}\} \cup (q_{\text{short}}, 0)$$

and accepting ones as:

$$\{(q_a, \infty) \mid q_a \text{ is an accepting state of } \mathcal{A}\} \cup \{(q_{\text{short}}, l) \mid 0 \leqslant l \leqslant n\}$$

We see that $L(\mathcal{A}') = L(\mathcal{A}) \cup \{u \in \Sigma^* : |u| \leqslant n\}$ and automaton $\mathcal{A}'$ is unambiguous because for words u of length at most n the only accepting run uses states $(q_{\text{short}}, 0), \ldots, (q_{\text{short}}, |u|)$ and for longer words the only accepting run corresponds to the accepting run of automaton $\mathcal{A}$.

Proof of lemma. The proof is fully analogous to the one, that if $w \in \Sigma^*$ is the shortest word such that $w \notin L(\mathcal{A})$, and automaton $\mathcal{A}$ is unambiguous with n states, then $|w| \leqslant n$.

Let us recall that for a given word $w = a_1 \ldots a_k$ we defined a $(k+1) \times (k+1)$ zero-one matrix M such that $M_{ij} = 1$ if and only if word $a_1 a_2 \ldots a_i \cdot a_{j+1} a_{j+2} \ldots a_k$ belongs to language $L(\mathcal{A})$. Then, it was shown that $k \leqslant \text{rank}(M) \leqslant n$, and thus $|w| = k \leqslant n$. The only difference from the original proof is that now matrix $J - M$ is lower-triangular (not upper triangular as before) because word w is the longest one such that $w \notin L(\mathcal{A})$.