

Problem 1: Definite Automata (30 pts.)**Background**

Finite state machines are often called memoryless devices since they do not have access to any kind of RAM memory. On the other hand, even a DFA can remember certain inputs unboundedly long. Here is a type of finite state machine that truly forgets.

An accessible DFA is *definite* if for some $r \geq 0$, all words of length at least r and all states p and q : $\delta(p, x) = \delta(q, x)$. Thus, in a definite DFA, acceptance of a sufficiently long input depends solely on the suffix of length r of the input. A regular language is definite if it can be accepted by a definite DFA.

Task

- A. Show that $L_{a,-3}$, the language of all words with an a in position -3 , is definite.
- B. Let $A \subseteq \Sigma^*$ be any finite language. Show that $\Sigma^* A$ is definite.
- C. Develop an algorithm that tests if a DFA is definite. Make sure to explain why your algorithm is correct.
- D. What is the complexity of your algorithm?

Problem 2: Right Quotients (30 pts.)**Background**

Left quotients are defined by removing a prefix from a word. Needless to say, we can also define *right quotients* by removing a suffix:

$$L/x = \{y \in \Sigma^* \mid yx \in L\}$$

Algebraically, these right quotients behave very much like their left cousins.

Left quotients are directly related to behaviors and can be used to decompose a regular language; one might wonder if regular languages can also be described in some interesting way in terms of right quotients. The answer is yes, but things become a bit more complicated.

In the following, assume $M = \langle Q, \Sigma, \delta; q_0, F \rangle$ is the minimal DFA accepting some language L . Fix some state p and, for all states $q \neq p$, choose a witness $w_q \in \Sigma^*$ such that $\delta(p, w_q) \in F \iff \delta(q, w_q) \notin F$ (which witness exists since M is minimal).

Task

- A. Show that for any Boolean operation \oplus we have $(L \oplus K)/x = L/x \oplus K/x$.
- B. Let $K \subseteq L$ be the language accepted by the automaton $\langle Q, \Sigma, \delta; q_0, \{p\} \rangle$. Show that $K = \bigcap_{q \neq p} R_q$ where $R_q = L/w_q$ whenever $\delta(p, w_q) \in F$, and $R_q = \Sigma^* - L/w_q$ otherwise ($\delta(p, w_q) \notin F$).
- C. Correspondingly, write L as a Boolean combination of right quotients of L .

Problem 3: The Un-Equal Language (40 pts.)

Background

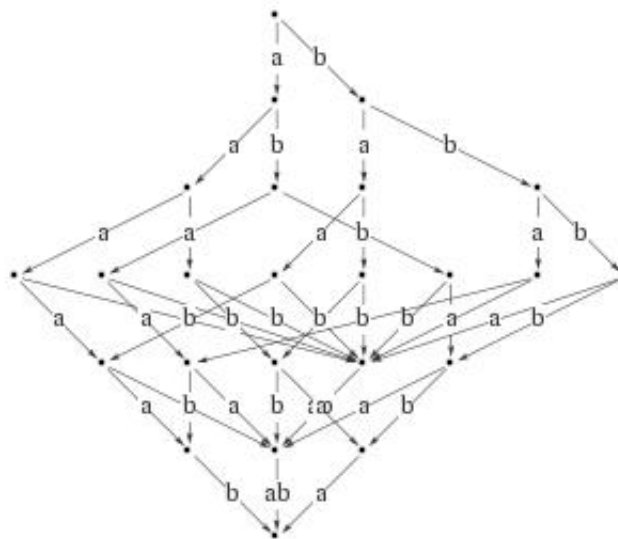
Consider the language of all strings of length $2k$ that are not of the form uu :

$$L_k = \{uv \in \{a, b\}^* \mid |u| = |v| = k, u \neq v\}.$$

These languages are finite, hence trivially regular. The following table shows the state complexity of L_k up to $k = 6$.

k	1	2	3	4	5	6
sc	5	12	25	50	99	196

The minimal DFA for L_3 looks like so (careful, some of the labels are printed on top of each other):



The sink has been omitted, the top state is initial and the bottom state final.

Task

- A. Determine all quotients for L_2 .
- B. Generalize. In particular explain the diagram for L_3 .
- C. Determine the state complexity of L_k .

Comment

From the diagram and the table it is not hard to conjecture a reasonable closed form for the state complexity. For a proof one can exploit the description of the minimal DFA in terms of quotients.