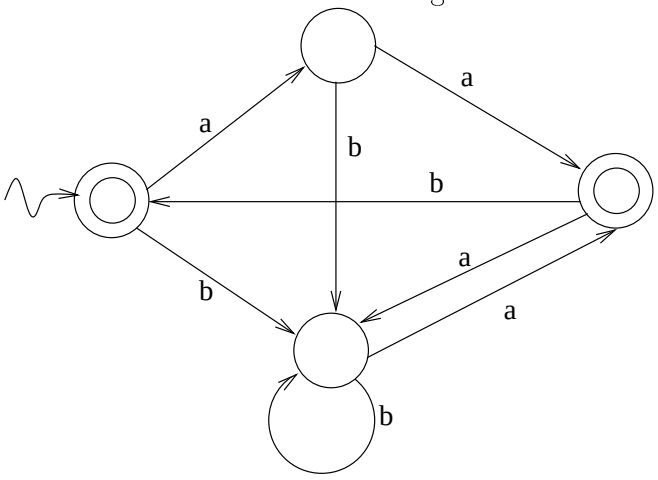
Final #1 - Questions and Solutions

1. Let $f : \{a, b\}^* \longrightarrow \{a, b\}^*$ be the function adding to each word starting with a another a in the beginning, and to each word starting with b another b^2 in the beginning, that is:

$$f(w) = \begin{cases} \varepsilon, & w = \varepsilon, \\ aw, & w = aw_1,\ w_1 \in \{a, b\}^*, \\ b^2 w, & w = bw_1,\ w_1 \in \{a, b\}^*. \end{cases}$$

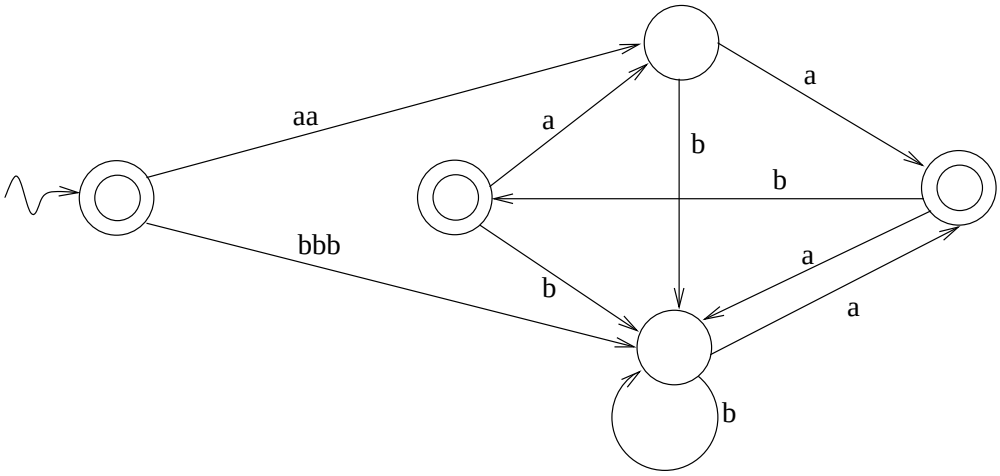
   (a) Prove that, if L is a regular language, then so is $f(L)$ in the following two ways:

      i. by constructing, given a finite automaton M accepting the language L, another finite automaton M' accepting $f(L)$. Define M' precisely, give a short explanation, and show how your construction works for the following automaton:



      Without loss of generality we may assume M to be a DFA, say $M = (Q, \Sigma, \delta, s, A)$. The automaton M' we construct may

well be an NFA. The idea is to replace the transition from s by reading a by the same transition, but reading a^2 instead (and similarly replace the transition from s by reading b to the same reading b^3). This does not work, however, as such changes will have effect any time the automaton returns to state s, whereas we want them to occur only at the initial stage. Therefore we add a new state s', which will serve as the initial state, and from it go to the states $\delta(s,a)$ and $\delta(s,b)$ by reading a^2 and b^3, respectively. Also note that to have $\varepsilon \in L(M')$ if and only if $\varepsilon \in L(M)$ we need to make the new initial state an accepting state if and only if s is an accepting state in M. Formally, $M' = (Q', \Sigma, \Delta, s', A')$, where:
$Q' = Q \cup \{s'\}, \quad (s' \notin Q),$
$\Delta = \{(p,\sigma,q) : \delta(p,\sigma) = q\} \cup \{(s', a^2, \delta(s,a)), (s', b^3, \delta(s,b))\},$
$A' = \begin{cases} A, & s \notin A, \\ A \cup \{s'\}, & s \in A. \end{cases}$
For the example in the question, the construction gives:



ii. using closure properties of the family of regular languages.

One verifies readily that:

$$f(L) = \{a\}\,(L \cap \{a\}\Sigma^*) \cup \{b^2\}\,(L \cap \{b\}\Sigma^*) \cup (L \cap \{\varepsilon\})\ .$$

As all the languages on the right hand side are regular, the

basic closure properties imply that $f(L)$ is regular as well.

(b) Show that, if L is a context-free language, then so is $f(L)$, by showing how, given a context-free grammar G accepting L, you can construct another context-free grammar G' accepting $f(L)$. You may assume for simplicity that the given grammar contains no rules of the form $V \to \varepsilon$ for $V \in N$. You do not have to define the new grammar formally; exemplify your construction on the following grammar:

$S \to AbB \mid BaA \mid b^2a^3$,

$A \to ABab \mid BAba \mid ab^2 \mid b^2a$,

$B \to BBA \mid AbbB \mid aba \mid bab$.

We need to add to any word which can be derived by the original rules an a or b^2 at the beginning, according to its initial letter. To explain the idea, assume first that in all grammatical rules of the form $S \to w$ the word w starts with either a or b, and that in no grammatical rule does the letter S appear on the right hand side. In this case we could simply change any rule of the form $S \to w$ to either $S \to aw$ or to $S \to b^2w$, according to whether w starts with an a or with a b, respectively. If the first assumption holds, but the second is violated, we could add a new initial letter S' and, for any rule of the form $S \to w$, either the rule $S' \to aw$ or $S' \to b^2w$, as before.

Now to the general case. The problem with a rule of the form $S \to w$, where $w \in N(N \cup \Sigma)^*$, is that we do not know in advance whether the word that will be obtained by the rules of G will start with an a or b. To overcome the problem we add to N a new letter for each letter of N. Denote the new letter corresponding to any $V \in N$ by V_0. The letter V_0 is almost the same as V, except that it stands at the beginning of the word to be derived, and thus eventually an a or b^2 will be added to its beginning. Thus, the rules corresponding to V are similar to those corresponding to V, but the first letter of their right hand side is changed to the version which must stand in the beginning of a word or an a or b^2 is added to it, depending on whether it starts with a letter of

N or of Σ, respectively. More formally, if $V \to Uw \in R$, where $U \in N$ and $w \in (N \cup \Sigma)^*$, then the rule $V_0 \to U_0 w$ is added to R'. If $V \to \sigma w \in R$ for some $\sigma \in \Sigma$ and $w \in (N \cup \Sigma)^*$, then the rule $V_0 \to f(\sigma)w$ is added to R'. In our example the rules are:

$S_0 \to A_0 bB \mid B_0 aA \mid b^4 a^3$,

$A_0 \to A_0 Bab \mid B_0 Aba \mid a^2 b^2 \mid b^4 a$,

$B_0 \to B_0 BA \mid A_0 bbB \mid a^2 ba \mid b^3 ab$,

$S \to AbB \mid BaA \mid b^2 a^3$,

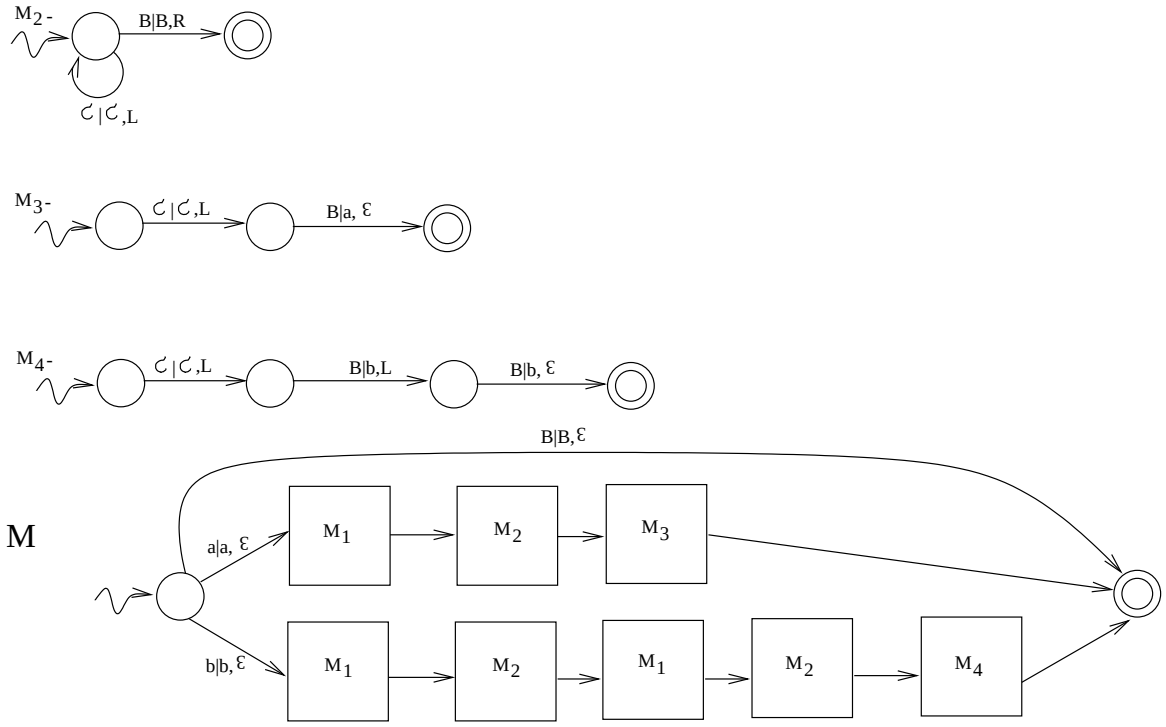
$A \to ABab \mid BAba \mid ab^2 \mid b^2 a$,

$B \to BBA \mid AbbB \mid aba \mid bab$.

Another possible solution is to add two new versions of each letter of N. Namely, if $V \in N$, then N' will include, in addition to V, the two letters V_a and V_b. These new letters are the same as the original V, except that their eventual derivation must start with an a or b, respectively. We also add a new initial symbol S' with the rules $S' \to aS_a \mid b^2 S_b$. The rules attached with V_a, say, are $V_a \to w$ if $V \to w \in R$ and $w \in a(N \cup \Sigma)^*$ and $V_a \to U_a w$ if $V \to Uw \in R$. This solution is more combersome than the first as the size of N is (more than) tripled rather than doubled.

(c) Construct a Turing machine computing the function f.

A possible solution is as follows. Let M_1 be the Turing machine described in class which moves all input one square to the right if it is non-empty, leaving the head at the rightmost square of the input, and finishing its action right on if the input is empty. Note that, if the word on the tape is $B^k w$ with $w \in (\Gamma - \{B\})^*$, and the head is at the leftmost character of w, then M_1 moves w another square to the right, so that the word on the tape is changed to $B^{k+1} w$. Let M_2 be a machine returning the head to the leftmost character of the input (in case a blank was added before the input), and let M_3 and M_4 be machines adding a or b^2, respectively, at the blank squares preceding the input. The following machine performs the required job.



2. Prove or disprove:

   (a) If $(L_n)_{n=1}^{\infty}$ is a sequence of regular languages, then the language $\cap_{n=1}^{\infty} L_n$ is Turing-acceptable.

   False. We know that there exist languages which are not Turing-acceptable. However, every language L is an intersection of countably many regular languages. In fact, take all languages of the form $\Sigma^* - \{w\}$ with $w \notin L$. The family is countable (since Σ^* is), consists of regular languages, and its intersection is exactly L.

   (b) Let L be a context-free language. Let L' be the language consisting of all words $w \in L$ for which there exists no decomposition $w = uvxyz$ with $vy \neq \varepsilon$ such that $uv^n xy^n z \in L$ for every $n \geq 0$. Then L' is regular.

   True. By the Pumping Lemma for context-free languages, the set of all such words w is finite, whence it is regular.

(c) Given a word $w \in \{a,b\}^*$, denote by $\bar{w}$ the word obtained from w by replacing each a by a b and each b by an a. (For example, if $w = bbab$ then $\bar{w} = aaba$.) Define a sequence of words $(w_n)_{n=1}^{\infty}$ inductively by:
$w_1 = a$,
$w_{n+1} = w_n \bar{w}_n, \qquad n \geq 1$.
Then $L = \{w_n : n \geq 1\}$ is context-free.

False. One verifies easily by induction that $|w_n| = 2^{n-1}$. Thus the set of lengths of all words in L is the set of powers of 2, which has unbounded gaps. For a context-free language, this set of lengths must be of bounded gaps (proof as in the case of regular languages).