Introduction to Formal Language, Fall 2017	23-Mar-2017 (Thursday)

Homework 2 Solutions

Instructor: Prof. Wen-Guey Tzeng *Scribe: Yi-Ruei Chen, Amir Rezapour*

1. Find all strings in $L((a+b)^*b(a+ab)^*)$ of length less than four.
Answer.

The strings with length 1: $\{\lambda b\lambda\} = \{b\}$;
The strings with length 2: $\{ab\lambda, bb\lambda, \lambda ba\} = \{ab, bb, ba\}$;
The strings with length 3: $\{(aa)b\lambda, (ab)b\lambda, (ba)b\lambda, (bb)b\lambda, (a)b(a), (b)b(a), \lambda b(ab)\} = \{aab, abb, bab, bbb, aba, bba, bab\}$. □

2. Find a regular expression for the set $\{a^n b^m : (n+m)\ is\ odd\}$.
Answer.

There are two cases:

- n is even and m is odd: $(aa)^*b(bb)^*$;
- n is odd and m is even: $a(aa)^*(bb)^*$;

Thus, a regular expression for the set $\{a^n b^m : (n+m)\ is\ odd\}$ is $(aa)^*b(bb)^* + a(aa)^*(bb)^*$. □

3. Give regular expression for the complement of $L_1 = \{a^n b^m, n \geq 3, m \leq 4\}$.
Answer.

$\overline{L}_1 = \overline{\{a^n b^m, n \geq 3, m \leq 4\}} = \{a^n b^m, n < 3\} \cup \{a^n b^m, m > 4\}$.
The regular expression for $\{a^n b^m, n < 3\}$ is $b^* + ab^* + aab^*$ and the regular expression for $\{a^n b^m, m > 4\}$ is $a^*bbbbbb^*$.
Thus, the regular expression for $\overline{L}_1$ is $(b^* + ab^* + aab^*) + a^*bbbbbb^*$. □

4. Find a regular expression for $L = \{w \in \{0,1\}^* : w$ has exactly one pair of consecutive zeros.
Answer.

The cases for two occurrences of 00 are 000 and 0011^*00.
Thus, the regular expression for L is $000 + 0011^*00$. □

5. Find a regular expression over $\{0,1\}$ for the all strings not ending in 10.
Answer.

The cases for the desired regular expressions are $(0+1)^*00$, $(0+1)^*01$, and $(0+1)^*11$.
The regular expression over $\{0,1\}$ for the all strings not ending in 10 is $(0+1)^*(00+01+11)$. □

6. Determine whether or not the following claim is true for all regular expressions r_1 and r_2. The symbol $\equiv$ stands for equivalence regular expressions in the sense that both expressions denote the same language.

   (a) $(r_1^*)^* \equiv r_1^*$.
   (b) $r_1^*(r_1 + r_2)^* \equiv (r_1 + r_2)^*$.
   (c) $(r_1 + r_2)^* \equiv (r_1 r_2)^*$.
   (d) $(r_1 r_2)^* \equiv r_1^* r_2^*$.

   Answer.

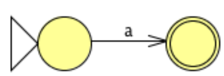
   (a) Yes. $\because L((r_1^*)^*) = (L(r_1^*))^* = ((L(r_1))^*)^* = (L(r_1))^* = L(r_1^*)$.
   (b) Yes. $\because$ Since $L(r_1^*(r_1 + r_2)^*) \subseteq L((r_1 + r_2)^*(r_1 + r_2)^*) = L((r_1 + r_2)^*)$ and $L((r_1 + r_2)^*) = L(\lambda(r_1 + r_2)^*) \subseteq L(r_1^*(r_1 + r_2)^*)$, they are equivalent.
   (c) No. $\because (r_1 + r_2)^* = (\lambda + r_1 + r_2 + r_1 r_2 + \dots)^*$ and $(r_1 r_2)^* = (\lambda + r_1 r_2 + r_1 r_2 r_1 r_2 + \dots)^*$. Therefore, $(r_1 r_2)^* \subset (r_1 + r_2)^*$ but $(r_1 + r_2)^* \not\subset (r_1 r_2)^*$.
   (d) No. $\because (r_1 r_2)^* = (\lambda + r_1 r_2 + r_1 r_2 r_1 r_2 + \dots)^*$ and $r_1^* r_2^* = (\lambda + r_1 + r_2 + r_1 r_1 + \dots, r_1^n r_2^m)^*$.

   □

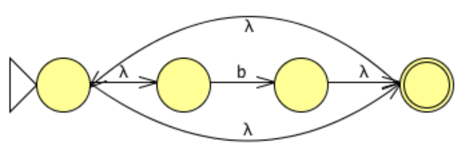
7. Use the construction in Theorem 3.1 to find an nfa that accepts the language $L(ab^*aa + bba^*ab)$.

   Answer.

   By Theorem 3.1, the automata for $L(a)$ is

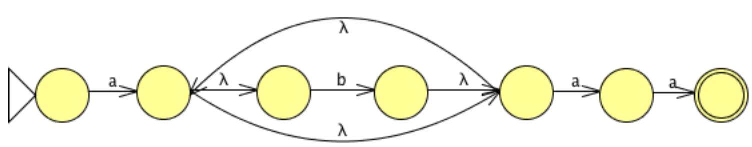


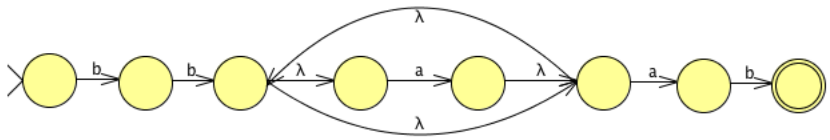
   By Theorem 3.1, the automata for $L(a^*)$ is

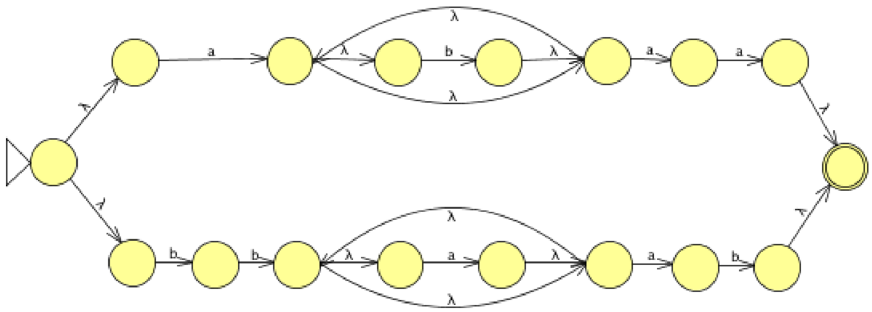


   The automata for $L(b)$ and $L(b^*)$ can be constructed in a similar way.

   Then by Theorem 3.1, the automata for $L(ab^*aa)$ is





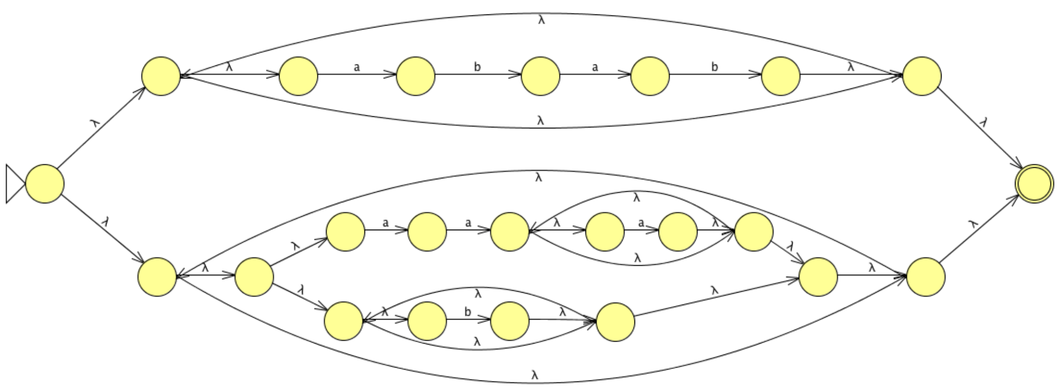


Then by Theorem 3.1, the automata for $L(bba^*ab)$ is

Thus, by Theorem 3.1, the automata for $L(ab^*aa + bba^*ab)$ is $\square$

8. Find an nfa that accepts the language $L((abab)^* + (aaa^* + b)^*)$.
Answer.

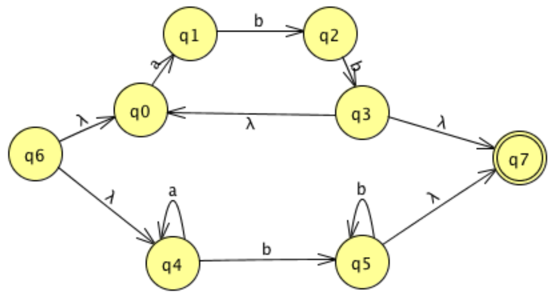
Similar to the steps in Question 7, an nfa that accepts the language $L((abab)^* + (aaa^* + b)^*)$ is as follows. $\square$



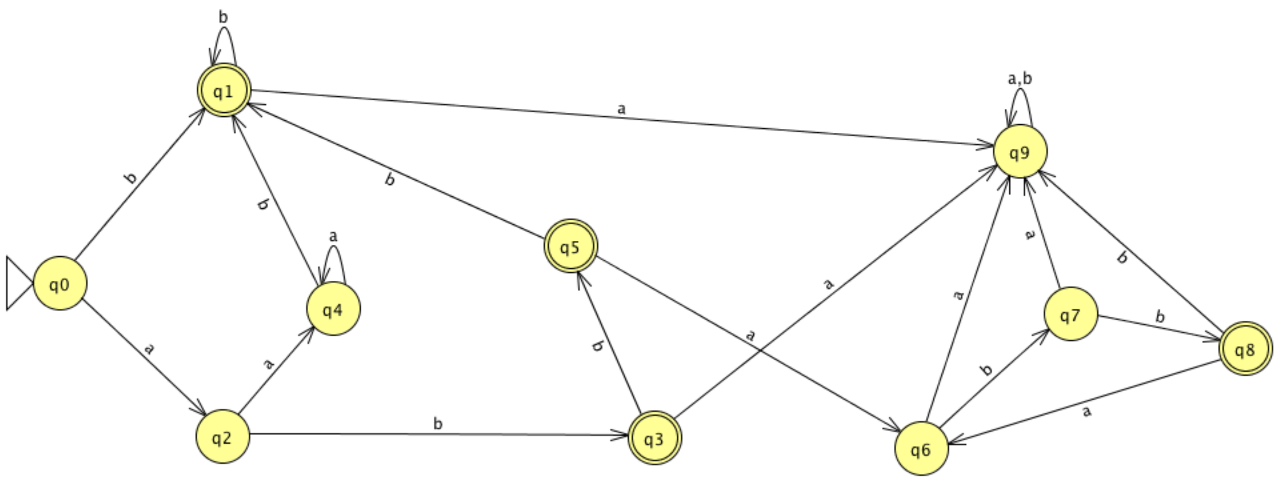
9. Find the minimal dfa that accepts $L(abb)^* \cup L(a^*bb^*)$.

Answer.

The following is an nfa that accepts $L(abb)^* \cup L(a^*bb^*)$.



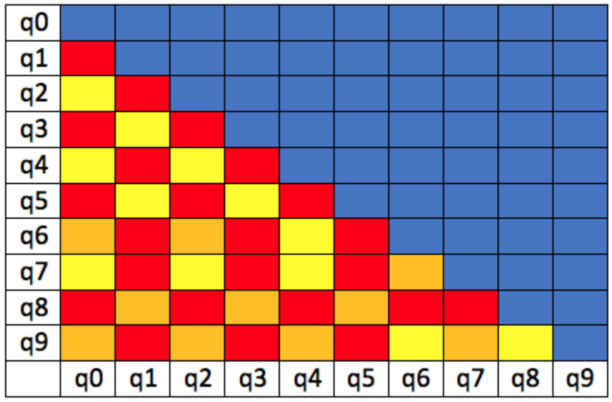
The following is the corresponding dfa that accepts $L(abb)^* \cup L(a^*bb^*)$.



Using Theorem 2.4 the corresponding minimized DFA is as follows. As shown in the table, in the first iteration (marked in red), we mark distinguishable states. For example, q_1 and q_0 are distinguishable since q_1 is final and q_0 is non-final state.

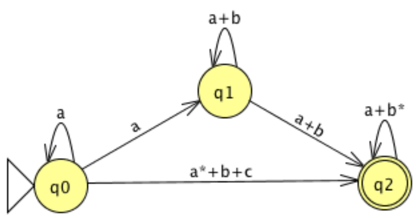
Next, we iterate over the remaining parts and check if they are distinguishable or not. For example, $\delta(q_2, a) = q_4 \notin F$,$\delta(q_0, a) = q_2 \notin F$ and $\delta(q_2, b) = q_3 \in F$,$\delta(q_0, b) = q_1 \in F$, hence so far they are indistinguishable. On the other hand, since $\delta(q_6, a) = q_9 \notin F$,$\delta(q_0, a) = q_2 \notin F$ and $\delta(q_6, b) = q_7 \notin F$,$\delta(q_0, b) = q_1 \in F$, q_0 and q_6 are distinguishable and marked with orange.

In the third iteration (marked in yellow), For all pairs (p, q) and $a \in \Sigma$, compute $\delta(p, a) = p_a$ and $\delta(q, a) = q_a$. If (p_a, q_a) is distinguishable, then mark (p, q) as distinguishable. For example, (q_2, q_7) are distinguishable since $(\delta(q_2, a) = q_4, \delta(q_7, a) = q_9)$ and (q_5, q_9) are distinguishable.

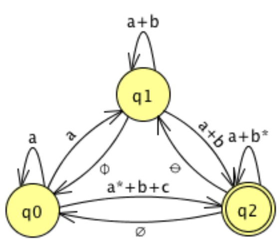


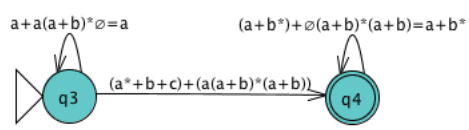
Finally, since all of the states are distinguishable, our designed dfa is already minimized. □

10. What language is accepted by the following automata.

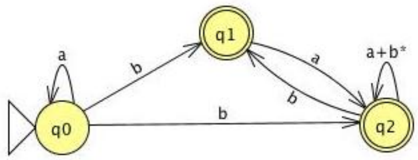


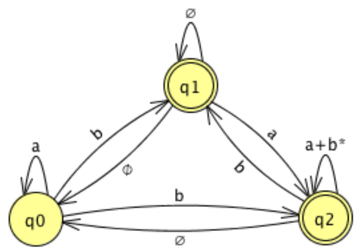
Answer. We first convert the given nfa to a complete GTG in the left-hand-side figure. Next, we reduce the state q_1 to obtain a two states one(Right-hand-side). Finally, we obtain a regular expression: $(a^*((a^*+b+c)+(a(a+b)^*(a+b)))+(a+b^*))$ □

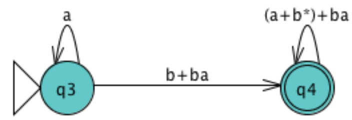




11. Find regular expression for the language accepted by the following automata.
Answer. We first convert the given nfa to a complete GTG in the left-hand-side figure. Next, we reduce the state q_1 to obtain a two states one(Right-hand-side). Finally, we obtain a regular expression: $(a^*(b+ba)((a+b^*)+ba)^*)$ □





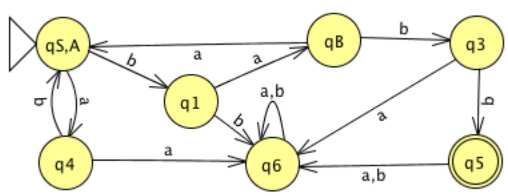


12. Write a regular expression for the set of all C real numbers.
Answer.
Let $d = 0+1+2+3+4+5+6+7+8+9$.
A regular expression for the set of all C real numbers is $r = ('+' +' -' + \lambda)dd^*(.dd* + \lambda)(Exp('+' +' -' + \lambda)dd^* + \lambda)$ □

13. Construct a dfa that accepts the language generated by the grammar

$$S \to abS|A,$$
$$A \to baB,$$
$$B \to aA|bb.$$

The dfa is constructed as follows, where q_x corresponds to variable $x, x \in \{S, A, B\}$.
Answer.

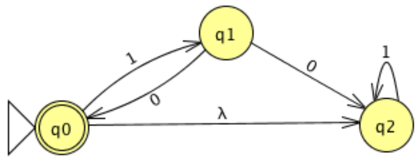


□

14. Construct right- and left-linear grammars for the language $L = \{a^n b^m : n \geq 3, m \geq 2\}$.
Answer.
The right-grammar $G_R = (S, A, B, \{a, b\}, S, P)$ with productions $S \to aaaA$, $A \to aA|bB$, $B \to bB|b$.

The left-grammar $G_L = (S, A, B, \{a, b\}, S, P)$ with productions $S \to Bbb$, $B \to Bb|Aaa$, $A \to Aa|a$. □

15. Use the construction suggested by the above exercises to construct a left-linear grammar for the nfa bellow.



Answer.

Notice that the language of the nfa is $L(M) = \{(10)^n : n \geq 0\}$. Therefore, the left-grammar $G_L = (S, \{0,1\}, S, P)$ with productions $S \to \lambda | S10$.

□

16. User the construction in Theorem 4.1 to find nfa that accept $L = ((ab)^*a^*) \cap L(baa^*)$.
Answer.

Let $L_1 = L((ab)^*a^*)$ and $L_2 = L(baa^*)$. We begin by designing a nfa for M_1 and M_2 as follows. Next, we simultaneously run $M_1 \times M_2$ to find an nfa $L_1 \cap L_2$. Since

there is no common transition from the initial states of M_1 and M_2, $L = \{\}$. That is, $L_1 = L((ab)^*a^*)$ and $L_2 = L(baa^*)$ have no intersection.

□

17. The *symmetric difference* of two sets S_1 and S_2 is defined as

$$S_1 \ominus S_2 = \{x : x \in S_1 \text{ or } x \in S_2, \text{ but } x \text{ is not in both } S_1 \text{ and } S_2\}.$$

Show that the family of regular languages is closed under symmetric difference.
Answer.

From the definition of symmetric difference of two sets, we have that $S_1 \ominus S_2 = (S_1 \cap \overline{S}_2) \cup (S_2 \cap \overline{S}_1)$. Because of the closure of regular languages under intersection ($\cap$), complementation ($\overline{L}$), and union ($\cup$), the family of regular languages is closed under symmetric difference. □

18. The tail of a language is defined as the set of all suffices of its strings, that is,

$$tail(L) = \{y : xy \in L \ for \ some \ x \in \Sigma^*\} \tag{1}$$

Show that if L is regular, so is $tail(L)$.
Answer.

Assume that dfa $M = (Q, \Sigma, \delta, q_0, F)$ accepts L and every state q in Q is reachable from q_0.

For every state q, there is string x with $\delta^*(q_0, x) = q$ (string x takes M from q_0 to q). Therefore, any string that takes M from q to a final state is in $tail(L)$ since xy is accepted with $\delta^*(q_0, xy) = \delta^*(\delta^*(q_0, x), y) \in F$. Thus, we add a new initial state q_0' and λ-transitions from q_0' to every state in M to form a new nfa M' so that $L(M') = tail(L)$, where M' is formally constructed as: M'=$(Q \cup \{q_0'\}, \Sigma, \delta', q_0', F)$, $\delta'(q_0', \lambda) = \{q : q \in Q\}$ and $\delta'(q, \sigma) = \delta(q, \sigma) \forall q \in Q \text{ and } \sigma \in \Sigma$.

□

19. For a string $a_1 a_2 \cdots a_n$ define the operation $shift$ as

$$shift(a_1 a_2 \cdots a_n) = a_2 \cdots a_n a_1.$$

From this, we can define the operation on a language as

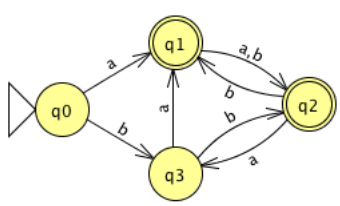
$$shift(L) = \{v : v = shift(w) \text{ for some } w \in L\}.$$

Show that the regularity is preserved under the $shift$ operation.
Answer.

Assume that the language L is given in DFA $M = (Q, \Sigma, \delta, q_0, F)$. We construct an NFA $N = (Q', \Sigma, \delta', q_s, \{q_f\})$ satisfies that $shift(L) = L(N)$ with $Q' = Q \times \Sigma \cup \{q_s, q_f\}$ and δ' as follows. (Note that $Q \times \Sigma = \{[q, \sigma] : q \in Q, \sigma \in \Sigma\}$, where $[q, \sigma]$ is a state for the first symbol of the string in L to be σ.)

- $\delta'(q_s, \lambda) = \{[\delta(q_0, \sigma), \sigma] : \sigma \in \Sigma\}$ (here we guess the first symbol to be σ, we will verify this guess "in the end" when the last symbol appears);
- $\delta'([q, \sigma], \sigma') = \{[\delta(q, \sigma'), \sigma]\}, \forall q \in Q, \forall \sigma, \sigma' \in \Sigma$;
- Add q_f to $\delta'([r, \sigma], \sigma), \forall r \in F, \forall \sigma \in \Sigma$;

For example, the following is a DFA M such that $L = L(M)$. The construction of the



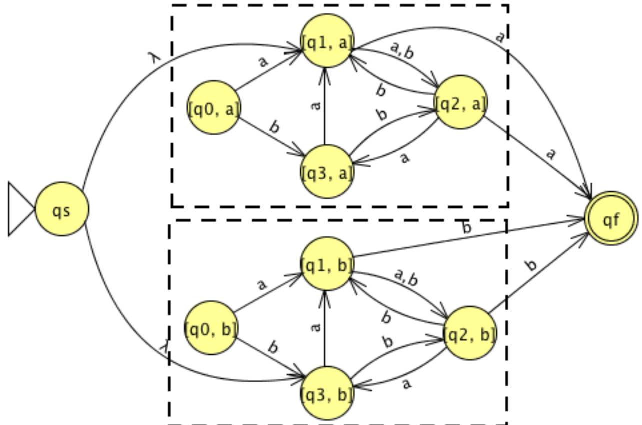
NFA N satisfies that $shift(L) = L(N)$ is shown as follows.

Thus, the regularity is preserved under the $shift$ operation.

□

20. Show that the following language is not regular. $L = \{a^n b^k c^n : n \geq 0, k \geq n\}$.
Answer.

Let m be the constant in the pumping lemma. We choose $w = a^m b^m c^m \in L$, $|w| \geq m$. For all possible x, y, z with $w = xyz$, $|xy| \leq m$, $|y| \geq 1$, there are following cases:



- Case 1: $x = a^{m-r}$, $y = a^r$, $z = b^\ell a^m$, $r \geq 1$. We let $i = 0$. $xy^0z = a^{m-r}b^\ell a^m \notin L$, because $m - r \neq m$.
- Case 2: no other cases.

Thus, L is not regular. $\square$

21. Show that the following language is not regular. $L = \{w : n_a(w) = n_b(w)\}$. Is L^* regular?
Answer.

Let m be the constant in the pumping lemma. We choose $w = a^m b^m \in L$, $|w| \geq m$. For all possible x, y, z with $w = xyz$, $|xy| \leq m$, $|y| \geq 1$, there are following cases:

- Case 1: $x = a^{m-r}$, $y = a^r$, $z = b^m$, $r \geq 1$. We let $i = 0$. $xy^0z = a^{m-r}b^m \notin L$, because $m - r \neq m$.
- Case 2: no other cases.

Thus, L is not regular. L^* is non-regular since $L^* = L$, which is shown to be non-regular. $\square$

22. Determine whether or not the following language on $\Sigma = \{a\}$ is regular

$$L = \{a^n : n = 2^k \text{ for some } k \geq 0\}.$$

Answer.

Let m be the constant in the pumping lemma. We choose $w = a^{2^m} \in L$, $|w| = 2^m \geq m$. For all possible x, y, z with $w = xyz$, $|xy| \leq m$, $|y| \geq 1$, there are following cases:

- Case 1: $x = a^r$, $y = a^s$, $z = a^{2^m - r - s}$, $r + s \leq m$, $s \geq 1$. We let $i = 2$. $xy^2z = a^r(b^s)^2z = a^{2^m+s} \notin L$, because

$$2^m < 2^m + s \leq 2^m + m < 2^m + 2^m = 2^{m+1}, 2^m + s \neq 2^k \text{ for any } k$$

- Case 2: no other cases.

Thus, L is not regular. $\square$

23. Make a conjecture whether or not the following language is regular. Then prove your conjecture.

$$L = \{a^n b^l a^k : n > 5, l > 3, k \leq l\}.$$

Answer.

Let m be the constant in the pumping lemma. We choose $w = a^6 b^m a^m \in L$, $m > 3$, $|w| = 2m + 6 \geq m$. For all possible x, y, z with $w = xyz$, $|xy| \leq m$, $|y| \geq 1$, there are following cases:

- Case 1: $x = a^r$, $y = a^s$, $z = a^{6-r-s} b^m a^m$, $r + s \leq 6$, $s \geq 1$. Let $i = 0$, $xy^0 z \notin L$ since less than 6 a's in the beginning.
- Case 2: $x = a^6 b^r$, $y = b^s$, $z = b^{m-r-s} a^m$, $s \geq 1$. Let $i = 0$, $xy^0 z = a^6 b^{m-s} a^m \notin L$ since the number of b's is less than the number of a's in the end.
- Case 3: y cannot contain both a's and b's since $xy^2 z$ is not in L.

Thus, L is not regular. $\square$

24. Let L_1 and L_2 be regular languages. Is the language $L = \{w : w \in L_1, w^R \in L_2\}$ necessarily regular?
Answer.

Yes, L is regular.
$L = \{w : w \in L_1, w^R \in L_2\} = \{w : w \in L_1\} \cap \{w : w^R \in L_2\}$, i.e., $L = L_1 \cap L_2^R$.
Because of the closure of regular languages under intersection and reverse (proved by Question 14 in HW1), L is regular. $\square$

25. Is the following language regular? $L = \{uww^R v : u, v, w \in \{a, b\}^+\}$
Answer.

You can choose ww^R to be in the form 00 and 11. Hence, for any given string you can check to see if the middle substring is in the form of 00 and 11. Therefore, L is regular and can be expressed as $(a + b)(a + b)^*(aa + bb)(a + b)^*(a + b)$. For example, $abab\underline{aa}bbba \in L$. $\square$