The University of Melbourne
School of Computing and Information Systems
COMP30026 Models of Computation

Selected Tutorial Solutions, Week 10

68. (a) $\{w \mid w$ begins with a `1` and ends with a `0`$\}$: $1(0 \cup 1)^*0$
   (b) $\{w \mid w$ contains the substring `0101`$\}$: $(0 \cup 1)^*0101(0 \cup 1)^*$
   (c) $\{w \mid w$ has length at least 3 and its third symbol is `0`$\}$: $(0 \cup 1)(0 \cup 1)0(0 \cup 1)^*$
   (d) $\{w \mid$ the length of w is at most 5$\}$: $(\epsilon \cup 0 \cup 1)(\epsilon \cup 0 \cup 1)(\epsilon \cup 0 \cup 1)(\epsilon \cup 0 \cup 1)(\epsilon \cup 0 \cup 1)$
   (e) $\{w \mid w$ is any string except `11` and `111`$\}$: $\epsilon \cup 1 \cup 11111^* \cup (0 \cup 1)^*0(0 \cup 1)^*$
   (f) $\{w \mid$ every odd position of w is a `1`$\}$: $(1(0 \cup 1))^*(\epsilon \cup 1)$
   (g) $\{w \mid w$ contains at least two `0`s and at most one `1`$\}$: $0^*(00 \cup 001 \cup 010 \cup 100)0^*$
   (h) $\{\epsilon, \mathtt{0}\}$: $\epsilon \cup 0$
   (i) The empty set: $\emptyset$
   (j) All strings except the empty string: $(0 \cup 1)(0 \cup 1)^*$

69. (a) We show that $A = \{\mathtt{0}^n\mathtt{1}^n\mathtt{2}^n \mid n \geq 0\}$ is not regular. Assume it is, and let p be the pumping length. Consider $s = 0^p1^p2^p \in A$. Since $|s| \geq p$, by the pumping lemma, s can be written $s = xyz$, so that $y \neq \epsilon$, $|xy| \leq p$, and $xy^iz \in A$ for all $i \geq 0$. But since $|xy| \leq p$, y must contains 0s only. Hence $xz \notin A$. Thus we have a contradiction, and we conclude that A is not regular.
   (b) We use the pumping lemma to show that B is not regular. Assume that it were. Let p be the pumping length, and consider $\mathtt{a}^{p+1}\mathtt{b}\mathtt{a}^p$. This string is in B and of length $2p+2$. By the pumping lemma, there are strings x, y, and z such that $\mathtt{a}^{p+1}\mathtt{b}\mathtt{a}^p = xyz$, with $y \neq \epsilon$, $|xy| \leq p$, and $xy^iz \in B$ for all $i \in \mathbb{N}$. By the first two conditions, y must be a non-empty string consisting of `a`s only. But then, pumping down, we get xz, in which the number of `a`s no longer is strictly larger than the number of `b`s. Hence we have a contradiction, and we conclude that B is not regular.
   (c) We want to show that $C = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is not a palindrome$\}$ is not regular. But since regular languages are closed under complement, it will suffice to show that $C^c = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome$\}$ is not regular. Assume that C^c *is* regular, and let p be the pumping length. Consider $a^pba^p \in C^c$. Since $|s| \geq p$, by the pumping lemma, s can be written $s = xyz$, so that $y \neq \epsilon$, $|xy| \leq p$, and $xy^iz \in C^c$ for all $i \geq 0$. But since $|xy| \leq p$, y must contains as only. Hence $xz \notin C^c$. We have a contradiction, and we conclude that C^c is not regular. Now if C was regular, C^c would be regular too. Hence C cannot be regular.

70. If A is regular then *suffix*(A) is regular. Namely, let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA for A. Assume every state in Q is *reachable* from q_0. Then we can turn D into an NFA N for *suffix*(A) by adding a new state q_{-1} which becomes the NFA's start state. For each state $q \in Q$, we add an epsilon transition from q_{-1} to q.

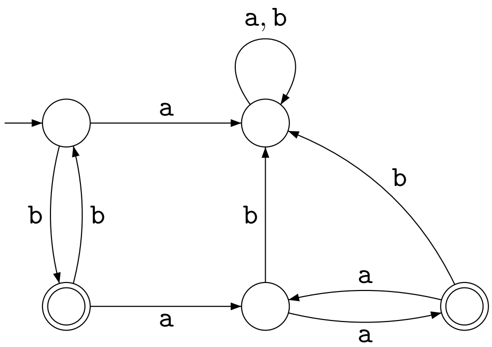
   That is, we define N to be $(Q, \Sigma, \delta', q_{-1}, F)$, with transition function

$$\delta'(q, x) = \begin{cases} \{\delta(q, x)\} & \text{for } q \in Q \text{ and } x \in \Sigma \\ Q & \text{for } q = q_{-1} \text{ and } x = \epsilon \\ \emptyset & \text{for } q \neq q_{-1} \text{ and } x = \epsilon \end{cases}$$

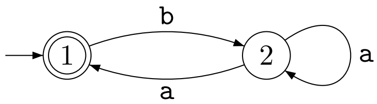
The restriction we assumed, that all of D's states are reachable, is not a severe one. It is easy to identify unreachable states and eliminate them (which of course does not change the language of the DFA). To see why we need to eliminate unreachable states before generating N in the suggested way, consider what happens to this DFA for $\{\epsilon\}$: $(\{q_0, q_1, q_2\}, \{\mathtt{a}\}, \delta, q_0, \{q_0\})$, where $\delta(q_0, \mathtt{a}) = \delta(q_1, \mathtt{a}) = q_1$ and $\delta(q_2, \mathtt{a}) = q_0$.

71. Note that a string in A must contain at least one `b`. Moreover, all `a`s (if there are any) must come after all `b`s, because otherwise we would find a substring `ab` somewhere. So a regular expression for A is $\mathtt{b}(\mathtt{bb})^*(\mathtt{aa})^*$.

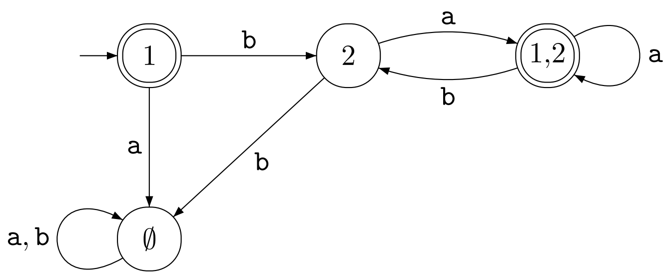
72. Here is a five-state DFA which recognises $\mathtt{b}(\mathtt{bb})^*(\mathtt{aa})^*$:



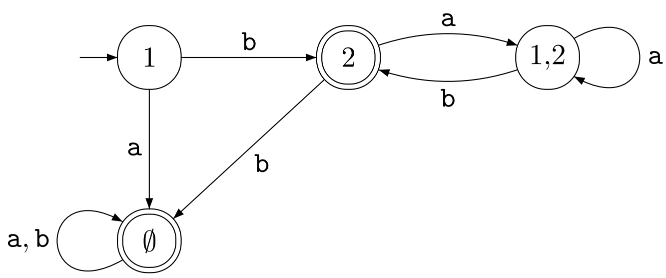
73. (a) Here is an NFA for $(\mathtt{ba}^*\mathtt{a})^*$:



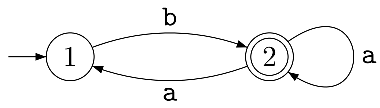
(b) Here is an equivalent DFA:



(c) It is easy to get a DFA for the complement:

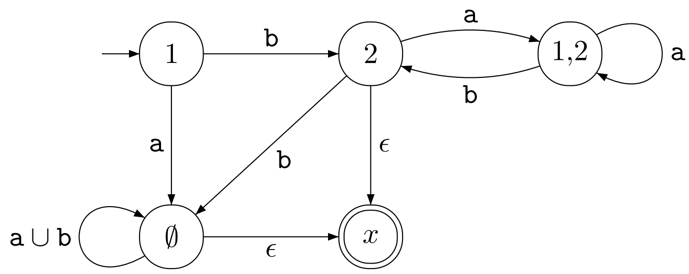


(d) It would be problematic to do the "complement trick" on the NFA, since it is only guaranteed to work on DFAs. We would get

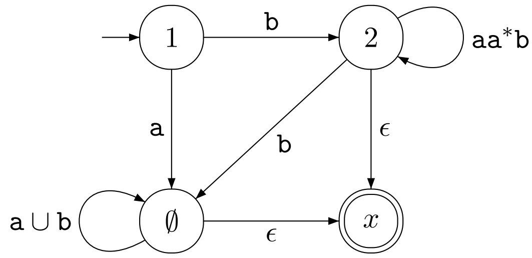


which accepts, for example, baa.

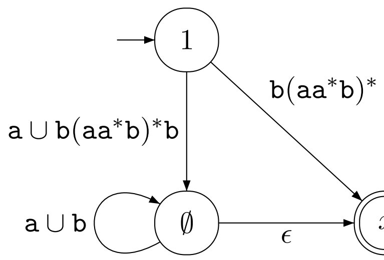
(e) Starting from the DFA that we found in (c), we make sure that we have just one accept state:



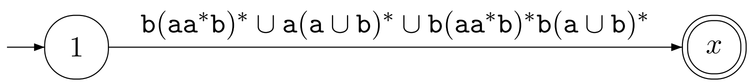
Let us first remove the state labeled 1,2:



Now we can remove state 2:



Finally, eliminating the state labelled $\emptyset$, we are left with:



The resulting regular expression can be read from that diagram.