1 The answer of Q3(a) (All strings in Σ^* with no more than three a's) was wrong. The file "Solution 1" has been updated in Tutorial webpage (https://comp3721tutorials.github.io)

The correct answer should be: $\boldsymbol{b^*(ab^* \cup b^*)(ab^* \cup b^*)(ab^* \cup b^*)}$.

2 About the empty set ϕ and empty word e

According to the definition in the lecture note:

ϕ is the empty language, i.e. the language contains no words.

e is the empty word containing no symbols, i.e. the word of zero length.

That is, ϕ is the language that contains nothing, which tallies with our consistent understanding about "empty set" while e is a word of zero length.

Note that since ϕ contains no element, so the concatenation of any other word or language with ϕ is ϕ (e.g. $\mathbf{b\phi = \phi}$, $\mathbf{L(\Sigma)L(\phi) = \phi}$) because there is no word in ϕ, resulting the whole string is empty.

That's why the original answer of Q3(a) is wrong. Recall that the original answer is $b^*(a \cup \phi)b^*(a \cup \phi)b^*(a \cup \phi)b^*$. However, $b^*\phi b^*\phi b^*\phi b^* = \phi \in b^*(a \cup \phi)b^*(a \cup \phi)b^*(a \cup \phi)b^*$. It is obviously incorrect.

$\boldsymbol{\phi^*} = \{\mathbf{e}\}$ because if we have 0 string from ϕ concatenated together, we will get the empty string.

3 The definition of regular expression

Thanks for the reminding of one student, the Wikipedia definition of regular expression (https://en.wikipedia.org/wiki/Regular_expression) is different with that in the lecture note. In Wikipedia, the empty string e is also the regular expression. There are divergences in the definition of regular expression from different sources. **In this course, we take the definition defined in the lecture notes**:

Inductive definition of *regular expressions* for languages over an alphabet Σ. A regular expression is a string over alphabet $\Sigma_1 = \Sigma \cup \{(,), \emptyset, \cup, ^*\}$.

1. $\emptyset$ and each $\sigma \in \Sigma$ are regular expressions.
2. If α and β are regular expressions, then

$$(\alpha\beta), (\alpha \cup \beta), \alpha^*$$

   are regular expressions.
3. Nothing else is a regular expression.

4 The detailed proof of the last problem:

Proof (follow the definition; **induction**):

i. $\forall L(a), a \in \Sigma \cup \phi$, if $a = \phi$, then $L(a) = \phi$, $\mathrm{Pref}(L(a)) = Pref(\phi) = \phi$;

   If $a \neq \phi$, then L(a)={a}, $\mathrm{Pref}(L(a)) = \phi$.

   ϕ is regular expression, so $\mathrm{Pref}(L(a))$ is regular.

ii. Suppose L1 and L2 are regular and have their Pref also regular. Now we need to prove the Pref of $L1 \cup L2, L1L2, (L1)^*$ are regular.

   a) $Pref(L1 \cup L2)$

   $\forall v_1 \in L1, v_2 \in L2,\ Pref(v_1 \cup v_2) = \mathrm{Pref}(v_1) \cup \mathrm{Pref}(v_2)$.

   Thus we see $Pref(L1 \cup L2) = Pref(L1) \cup Pref(L2)$

since Pref(L1) and Pref(L2) are regular language, $Pref(L1 \cup L2) =$ is regular.

b) $Pref(L1L2)$

$\forall v_1 \in L1, v_2 \in L2$, $Pref(v_1 v_2) = \{\text{w: w} \in \text{Pref}(v_1)\text{or w} = v_1 w_2, w_2 \in \text{Pref}(v_2)\}$.

Thus we see $Pref(L1L2) = Pref(L1) \cup (L1Pref(L2))$

since Pref(L1) , L1 and Pref(L2) are regular language, $Pref(L1L2)$ is regular.

c) $Pref((L1)^*)$

$\forall v_1, v_2 \dots, v_k \in L1$, $Pref(v_1 v_2 \dots v_k) = \{w: w = v_1 \dots v_{i-1} w_i, w_i \in \text{Pref}(v_i), \text{for i} = 1, \dots \text{k}\}$.

Thus we see $Pref((L1)^*) = (L1)^* Pref(L1)$

since L1 and Pref(L1) are regular, $Pref((L1)^*)$ is regular.

By induction, the statement is proved.

Note: the proof is a little different with what I told in tutorial. I think this one is easier to understand and more clear.

Acknowledgement: thank Yilei Wang for pointing out my previous mistakes.

If you have more questions about Tutorial 1, feel free to ask me. I’m willing to discuss with you. This file will be updated if I think there are other questions worthy of being added.