CPSC 313 — Hints for Question #1 in Tutorial Exercise #3

Recommendation: Please spend a bit of time trying to solve the problems in Question #1 ***before*** looking at the following hints — and try to use as few of these hints as possible! You will be better prepared for later assignments and tests if you can solve problems without hints like these.

That said, please ***do*** use them if you need a bit of help getting started.

In this question you were asked to design deterministic finite automata for each of the following languages. In all cases, $L \subseteq \Sigma^\star$ where $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

(a) $L = \{\lambda\}$

Hint: Keep it simple when you can! It is sufficient to remember whether any symbols have been seen yet.

(b) $L = \{\omega \in \Sigma^\star \mid \omega$ includes at most one $\mathtt{c}\}$

Hint: It might be tempting to try to remember the number of $\mathtt{c}$'s that have been seen so far. Unfortunately you would need an infinite number of states to do that.

Fortunately this is not necessary: It is sufficient to "stop counting" once a second $\mathtt{c}$ has been seen — so that it suffices to remember which of the following cases is applicable.

- No $\mathtt{c}$'s have been seen yet.
- Exactly one $\mathtt{c}$ has been seen so far.
- Two or more $\mathtt{c}$'s have already been seen.

(c) $L = \{\omega \in \Sigma^\star \mid \omega$ includes exactly one $\mathtt{c}\}$

See the answer for the previous question. Does anything need to be changed?

(d) $L = \{\omega \in \Sigma^\star \mid \mathtt{cc}$ is a substring of $\omega\}$

It is certainly necessary to remember whether the string that has been so far has $\mathtt{cc}$ as a substring — but you also need to remember a little bit more, if it does not, in order for transitions to be well defined: You also need to remember whether *part* of the string $\mathtt{cc}$ has been seen *at the end* of the string that you have already seen.

In particular, you should find that it is necessary — and sufficient — to keep track of the following cases.

- `cc` is not a substring of the string that has been seen so far, *and* this string does not end with a `c`.
- `cc` is not a substring of the string that has been seen so far, *but* this string *does* end with a `c`.
- `cc` *is* a substring of the string that has been seen so far.

(e) The set L of strings in $\Sigma^\star$ that include an `a`, with a `b` appearing eventually after that (but, possibly, with other symbols in between), and with another `a` appearing eventually after the `b` (possibly with other symbols in between, once again).

You need to keep track of how many of the events, suggested by the definition of this language, have already happened (in the required order). This should suggest the following cases.

- An `a` has not been seen yet.
- An `a` has been seen; `b` has not been seen after that.
- An `a` has been seen, and `b` was eventually seen after that. Another `a` has *not* yet been seen after *that*.
- An `a` has been seen. This was eventually followed by `b`, and *that* was eventually followed by another `a`.

(f) $L = \{\omega \in \Sigma^\star \mid \texttt{aba} \text{ is a substring of } \omega\}$

Try to modify the hint for part (d) so that it applies to the substring `aba` instead of the substring `cc`.

(g) $L = \{\omega \in \Sigma^\star \mid \text{the length of } \omega \text{ is divisible by } 4\}$

Once again, it might be tempting to try to remember the length of the string ω that has been seen so far — but this cannot be done because an infinite number of states would be required.

Fortunately, it is not necessary: You should be able to confirm that it is sufficient to keep track of the following cases.

- The length of ω is divisible by 4 — that is, $|\omega| \equiv 0 \pmod 4$.
- The length of ω is equal to $4k + 1$ for some integer k — that is, $|\omega| \equiv 1 \pmod 4$.
- The length of ω is equal to $4k + 2$ for some integer k — that is, $|\omega| \equiv 2 \pmod 4$.
- The length of ω is equal to $4k + 3$ for some integer k — that is, $|\omega| \equiv 3 \pmod 4$.