COSC 341 – Assignment 1

Solutions

1. *Let n be a positive integer, and let Σ be an alphabet containing k symbols. How many different DFAs are there over the alphabet Σ with state set $\{0, 1, 2, \ldots, n-1\}$ and initial state 0?*

Answer

The transition table of such a DFA has n rows (for the states) and k columns (for the symbols) and each entry is one of the n states. Since these entries can be filled independently, there are n^{kn} transition tables. Also, each state can be accepting or not, leading to 2^n further choices of status for the states, and a final answer of:

$$2^n \times n^{kn}.$$

---

2. *Let $k > 2$ be a positive integer, and let Σ be the alphabet $\{0, 1, 2, \ldots, k-1\}$. Consider the language $L \subseteq \Sigma^*$ consisting of all those strings such that no consecutive pair of letters differ by 1 modulo k (so $k-1$ cannot be immediately followed, nor preceded by 0 or $k-2$).*

- *Is this language regular?*

Answer

Yes, this language is regular. For $k = 1$ it consists either of $\{\epsilon, 0\}$ or 0^* (depending on how we interpret the condition 'differ by 1 modulo 1'). For $k \geqslant 2$ we can describe an automaton with $k+2$ states that accepts it as follows:

- The states are labelled S (for start), 0, 1, ..., $k-1$, and G (for garbage).
- The start state is S and all states except G are accepting.
- The transition from S on letter j is to state j. All transitions from G lead to G.
- The transitions from j on letter t are to state t unless t differs from j by 1 modulo k in which case it is to G.

This automaton clearly accepts the given language.

Other arguments are possible. For instance for each pair of letters a, b that *do* differ by 1 we know the language Has-ab is regular. Therefore, the complement of each of these languages is regular, as is the intersection of all those complements – and that's the language specified.

Or we could apply Myhill-Nerode. For this language L, if $w \notin L$ its suffix language is empty. If $w = \epsilon$ its suffix language is L. Finally, if $w \in L$ but is not ϵ then its suffix language is all those words in L that do not begin with the characters that the final character of w forbids. So there are $k+2$ different suffix languages, and hence the language is regular.

- *For a positive integer n, how many strings of length n are there in L?*

Answer

The cases $k \leqslant 2$ are slightly exceptional. As noted above, for $k = 1$ the answer is either "1 for $n = 0, 1$, 0 thereafter" or "1 for all n" (depending on how we interpret the condition 'differ by 1 modulo 1'). For $k = 2$ the answer is 2 for all n (since we can choose the first character but must repeat it thereafter).

For $k \geqslant 3$ we are free to choose the first character (k possibilities) and then at each subsequent choice there are 2 forbidden characters, so we have $k - 2$ possibilities. For a total of n characters this gives

$$k \times (k-2)^{n-1}$$

words in the language.

---

3. *Let L be the language over $\Sigma = \{a, b\}$ of strings containing an even number of a's and not containing consecutive b's.*

   - Construct a DFA that accepts L.

Answer

The automaton will have five states: 0 (start state), $0b$, 1, $1b$, and G. The intended interpretations are as follows:

- In state 0 we have seen an even number of a's and the last character was not a b.
- In state $0b$ we have seen an even number of a's and the last character was a b.
- In state 1 we have seen an odd number of a's and the last character was not a b.
- In state $1b$ we have seen an odd number of a's and the last character was a b.
- In state G we have failed (we saw consecutive b's at some point)

The states 0 and $0b$ are accepting and the transitions are as follows:

	a	b
0	1	$0b$
$0b$	1	G
1	0	$1b$
$1b$	0	G
G	G	G

- *From the previous question, describe an NFA with the same set of states that accepts L^*.*

Answer

Add a single ϵ-transitions from $0b$ to 0. Optionally, make $0b$ not accepting. Additionally, for simplicity in the next part, we can delete the garbage state.

- *Convert that NFA to a DFA.*

Answer

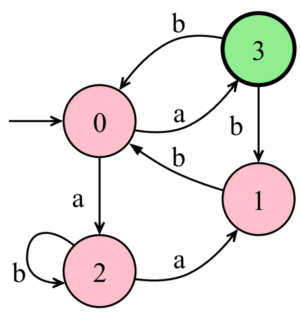
There's a single ϵ-transition (from $0b$ to 0) so we can use the original states (except $0b$) to denote their epsilon-closures, and use $00b$ to denote "either 0 or $0b$". This allows us to reconstruct a transition table (no other combined states arise, and we reintroduce a garbage state).

	a	b
0	1	$00b$
$00b$	1	$00b$
1	0	$1b$
$1b$	0	G
G	G	G

In fact, with a little thought we can see that this language is really "any pair of consecutive b's must be preceded by an even number of a's" which means that we don't really need to distinguish the 0 and $00b$ states. This can be seen from the table as well, since both are accepting and they have identical outgoing transitions.

---

4. *Consider the NFA below:*



Starting from a version of this NFA in standard form, illustrate the use of the state elimination technique to produce a regular expression for the language accepted by this NFA.

Answer

One answer: $(ab^*ab + a(b + bb))^*a$.

---

5. *Let $\Sigma = \{a, b\}$ and consider the language $L \subseteq \Sigma^*$ consisting of all those strings containing three consecutive a's. What are the suffix-equivalence classes for L?*

Answer

There are four different suffix languages for $w \in \Sigma^*$ with respect to L (and as we describe them, we will also specify those words that form the corresponding suffix-equivalence class):

- If $w \in L$ then $\mathrm{Suff}(w, L) = \Sigma^*$ (since we already have aaa so we can add absolutely anything).

- If $w \notin L$ but w ends in b or $w = \epsilon$ then $\mathrm{Suff}(w, L) = L$ (since we must append a word containing aaa to w to obtain an aaa).
- If $w \notin L$ but w ends in a but not aa then $\mathrm{Suff}(w, L) = aa\Sigma^* \cup L$ (since we can append anything beginning aa to get aaa or anything containing aaa).
- If $w \notin L$ but w ends in aa then $\mathrm{Suff}(w, L) = a\Sigma^* \cup L$ (since we can append anything beginning a to get aaa or anything containing aaa).