CS323 Assignment 1 - Answers

(For reference only. DO NOT distribute.)

Exercise 1: (1) appearance of illegal characters, (2) identifiers that start with digits

Note that spelling mistakes (e.g., misspelling "bar" as "bao") are generally not regarded as lexical errors. The examples listed on this page https://www.javatpoint.com/lexical-error are for reference only (may not be fully correct).

Exercise 2:

1. Yes, x can be s itself
2. It depends. To be a proper prefix of s, x must start with the first character in s. To be a proper suffix of s, x must end with the last character in s. If s's prefix is the same as its suffix (e.g., s is "aba"), then it is possible to find a string that is both a proper prefix and a proper suffix of s. However, if the prefix and suffix of s are different, it is not possible to find an x that satisfies our requirement.

Exercise 3:

1. $n - m + 1$
2. $\sum_{i=0}^{n} C_n^i = 2^n$

Exercise 4:
$digit \rightarrow [0 - 9]$
$nonZeroDigit \rightarrow [1 - 9]$
$countryCode \rightarrow 86$
$areaCode \rightarrow 755$
$phoneNumber \rightarrow countryCode - areaCode - nonZeroDigit\ digit^7$

Exercise 5:

L_1 is equivalent to L_2 and we can prove this by contradition (there may be other ways to prove this). Since L_1 and L_2 are essentially two sets, to prove that they are equivalent is basically to prove that (1) $L_1 \subseteq L_2$ and (2) $L_2 \subseteq L_1$.

Part 1: We first prove that $L_1 \subseteq L_2$.

Suppose that $L_1 \subseteq L_2$ does not hold, then we can find a string s such that $s \in L_1$ but $s \notin L_2$. Without loss of generality, let's assume that $|s| = n$ and we use (i, j) to denote a substring

of s, where i indicates the index of the starting character and j indicates the index of the ending character. Here, the index starts from 1, $1 \leq i, j \leq n$, and $i \leq j$.

If s is not in L_2, then its substring $(1, n-1)$ is also not in L_2, because if $(1, n-1)$ is in L_2, then obviously s is also in L_2 according to the language definition. Next, we can show that substrings $(1, n-2)$, $\ldots$, $(1, 1)$ are not in L_2. This is impossible since $(1, 1)$ is either the string "0" or "1", which must be in L_2.

So $L_1 \subseteq L_2$ must hold.

Part 2: We then prove that $L_2 \subseteq L_1$.

Suppose that $L_2 \subseteq L_1$ does not hold, then we can find a string s such tat $s \in L_2$ but $s \notin L_1$. Without loss of generality, let's assume that $|s| = n$ and we use (i, j) to denote a substring of s, where i indicates the index of the starting character and j indicates the index of the ending character. Here, the index starts from 1, $1 \leq i, j \leq n$, and $i \leq j$.

The language $L(0^*1^*)$ is equivalent to the language $L(0|1|0^*1^*)$, i.e., adding two strings "0" and "1" does not change the set. Then $L((0^*1^*)^*)$ is equivalent to $L((0|1|0^*1^*)^*)$.

If s is not in L_1, then its substring $(1, n-1)$ is not in L_1, because if $(1, n-1)$ is in L_1, then obviously s is also in L_1 according to the language definition. Next, we can show that substring $(1, n-2)$, $\ldots$, $(1, 1)$ are not in L_1. This is impossible since $(1, 1)$ is either the string "0" or "1", which must be in L_1.

So $L_2 \subseteq L_1$ must hold.

Exercise 6:

The transition diagram:

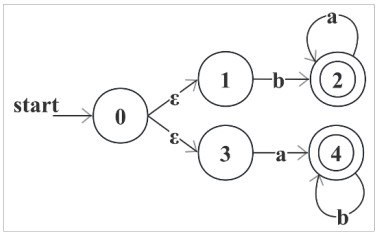

Figure 1: state transition diagram example

This transition diagram can not recognize the string baab. The sequence of state transitions can be shown as follow (we cannot find a sequence of state transitions starting from 0 and ending at an accepting state to consume the string baab):

current state	input character	left string	next state
	$\widehat{}$(start)	baab$	0
0	ϵ	baab$	1
1	b	aab$	2
2	a	ab$	2
2	a	b$	2
2	b	$	null