CSC236 tutorial exercises #7
(sample solution)

Danny Heap

These exercises are meant to give you practice with regular expressions, see Section 7.2 of Course Notes.

1. Equivalence of regular expressions is discussed in Section 7.2.4 of the Course Notes, and direct proof of equivalence in Section 7.2.5. Prove or find a counterexample:

\[ SR \equiv RS \Rightarrow S \equiv R \]

Does it make any difference if we insist that neither \( S \) nor \( R \) are \( \{, \} \)?

**Solution:** One counterexample is \( R = \varepsilon \) and \( S = (0 + 1)^* \). Then \( L(SR) = L(S) = L(RS) \). Another example is \( R = 0 \) and \( S = 00 \) (each language consists of a single string). Then \( RS = SR = 000 \).

2. Let \( L \) be the set of strings over \{0,1\} that begin and end with the same bit. Devise a regular expression, \( R \) that denotes \( L \), and prove that your regular expression is correct (see Course Notes page 194–195 for a related example).

**Solution:** First, note that \( \varepsilon \) is not in \( L \), since it doesn’t begin or end with a bit, different or the same as anything (if \( L \) has been defined as the set of strings that don’t begin with different bits, we’d have a different kettle of strings). Our regular expression has to account for single-symbol strings, where the first character is the last character, so here’s an RE:

\[ R = 0 + 1 + 0(0 + 1)^*0 + 1(1 + 0)^*1 \]

Now, I prove that \( L(R) = L \), in other words \( L(R) \subseteq L \) and \( L \subseteq L(\varepsilon) \).

**Proof:** Suppose \( s \) is an arbitrary string in \( L(R) \). Since \( L(R) \) is a union of four smaller languages, I consider those cases:

**Case** \( s \in L(0) \) or \( s \in L(1) \): Then either \( s = 0 \), or \( s = 1 \), and both of these begin and end with the same bit, 0 or 1, respectively

**Case** \( s \in L(0(0 + 1)^*0) \) or \( s \in L(1(1 + 0)^*1) \): Without loss of generality\(^1\) assume \( s \in L(0(0 + 1)^*0) \). Then, by definition of the language denoted by \( R \), \( s \) is of the form \( tuv \), where \( t \in L(0) \), \( u \in L((0 + 1)^*) \), and \( v \in L(0) \). Thus \( s \) begins and ends with the same bit: 0

In all possible cases, \( s \) begins and ends with the same bit, so \( s \in L \). Since \( s \) was chosen to be an arbitrary element of \( L(R) \), this means that every element of \( L(R) \) is also in \( L \), in other words \( L(R) \subseteq L \).

Now suppose that \( s’ \) is an arbitrary element of \( L \). I consider the cases where \( |s’| \leq 1 \) and \( |s’| > 1 \) separately

---

\(^1\)Since the same argument works for \( s \in L(1(1 + 0)^*1) \), simply interchanging 0s and 1s throughout
Case $|s'| \leq 1$: In this case $|s'| = 1$, since the lone binary string of length 0 has neither a starting nor ending bit to be the same as anything. The binary strings of length 1 are 0 and 1, and they belong, respectively, to $L(0)$ and $L(1)$, and each of these are subsets of $L(R)$, by inspection, so $s' \in L(R)$.

Case $|s'| > 1$: In this case, the starting and ending bits of $s'$ are the same, but at distinct indices in $s'$, separated by 0 or more symbols from $\{0, 1\}$, so $s' = tut$, where $t \in L(0 + 1)$, and $u$ is an arbitrary binary string over $\{0, 1\}$, that is $u \in L((0 + 1)^*)$. Without loss of generality, assume $t \in L(0)$, since the same argument follows if I interchange 0s for 1s. In this case, $s' \in L(0)L((0 + 1)^*)L(0) \subseteq L(R)$. So $s' \in L(R)$.

In both possible cases, $s' \in L(R)$, and since $s'$ was chosen arbitrarily, every string of $L$ is also a string of $L(R)$, so $L \subseteq L(R)$.

Since $L \subseteq L(R) \subseteq L$, this means $L = L(R)$.

---

2I assume assertion from page 193 of the Course Notes.