CMPS 130
Spring 2016
Homework Assignment 2 Solutions

Chapter 1 (p.34): 31, 33abc, 36ab, 44abcd (answer only, proof not necessary), 52, 63, 65ab, 66

1. Problem 1.31
   Show that for every language L, $LL^* = L^*$ if and only if $\lambda \in L$.

   Solution:
   We must prove two things: $LL^* = L^* \Rightarrow \lambda \in L$ and $\lambda \in L \Rightarrow LL^* = L^*$
   Proof of ($\Rightarrow$): Assume $LL^* = L^*$. We must show that $\lambda \in L$. Since $\lambda \in L^*$, there exists $x \in L$ and $y \in L^*$ such that $xy = \lambda$. Therefore $0 = |\lambda| = |xy| = |x| + |y|$, hence $|x| = |y| = 0$ and $x = y = \lambda$. In particular $\lambda = x \in L$.
   Proof of ($\Leftarrow$): Assume $\lambda \in L$. We must show that $LL^* = L^*$. First note that $LL^* \subseteq L^*$ is always true (whether or not $\lambda \in L$.) To see this suppose $x \in LL^*$. Then $x = yz$ where $y \in L$ and $z \in L^k$ for some $k \geq 0$. (Recall $L^* = \bigcup_{k=0}^{\infty} L^k$.) Thus $x \in L^{k+1}$, showing that $x \in L^*$, and hence $LL^* \subseteq L^*$, as claimed. It remains only to show $L^* \subseteq LL^*$ on the assumption $\lambda \in L$. Pick any $x \in L^*$. Then $x = \lambda x \in LL^*$. We've shown $LL^* \subseteq L^*$ and $L^* \subseteq LL^*$ when $\lambda \in L$, whence $LL^* = L^*$, as required.

2. Problem 1.33abc
   Let L_1 and L_2 be subsets of $\{a, b\}^*$.
   a. Show that if $L_1 \subseteq L_2$ then $L_1^* \subseteq L_1^*$.
   b. Show that $L_1^* \cup L_2^* \subseteq (L_1 \cup L_2)^*$.
   c. Give an example of two languages L_1 and L_2 such that $L_1^* \cup L_2^* \neq (L_1 \cup L_2)^*$.

   Solution:
   a. $L_1 \subseteq L_2 \Rightarrow L_1^* \subseteq L_2^*$
      Proof: Assume $L_1 \subseteq L_2$ and let $x \in L_1^*$. We must show that $x \in L_2^*$. Since $x \in L_1^*$ there exists an $n \geq 0$ such that $x \in L_1^n$. Therefore $x = x_1 x_2 x_3 \cdots x_n$ where each $x_i \in L_1$ for $1 \leq i \leq n$. Since $L_1 \subseteq L_2$ we actually have each $x_i \in L_2$ for $1 \leq i \leq n$. Therefore $x \in L_2^n$ and hence $x \in L_2^*$.
   b. $L_1^* \cup L_2^* \subseteq (L_1 \cup L_2)^*$
      Proof: Obviously $L_1 \subseteq L_1 \cup L_2$ and $L_2 \subseteq L_1 \cup L_2$. Part (a) now implies that $L_1^* \subseteq (L_1 \cup L_2)^*$ and $L_2^* \subseteq (L_1 \cup L_2)^*$. Therefore $L_1^* \cup L_2^* \subseteq (L_1 \cup L_2)^*$.
   c. $L_1^* \cup L_2^* \neq (L_1 \cup L_2)^*$
   d. **Example:** Let $L_1 = \{a\}$ and $L_2 = \{b\}$. Then $L_1 \cup L_2 = \{a, b\}$, $L_1^* = \{a^n \mid n \geq 0\}$ and $L_2^* = \{b^n \mid n \geq 0\}$. Therefore $L_1^* \cup L_2^* = \{\, x \in \{a, b\}^* \mid x = a^n \text{ or } x = b^n \text{ for some } n \geq 0\}$. Also we have $(L_1 \cup L_2)^* = \{a, b\}^* = \{\text{all strings on } a, b\}$. Clearly then $L_1^* \cup L_2^* \neq (L_1 \cup L_2)^*$ since for instance *ab* is in the right hand side, but not the left side.

3. Problem 1.36ab
   a. Consider the language L of all strings of *a*'s and *b*'s that do not end with *b* and do not contain the substring *bb*. Find a finite language S such that $L = S^*$.
   b. Show that there is no language S such that S^* is the language of all *a*'s and *b*'s that do not contain the substring *bb*.

Solution:

a. Let $S = \{a, ba\}$. Then it is claimed that $S^* = L$, where L is the set of all strings $x \in \{a, b\}^*$ such that (i) x does not end in b, and (ii) x does not contain the substring bb.

Proof: We must show that both $S^* \subseteq L$ and $L \subseteq S^*$.

Proof of $S^* \subseteq L$: Since $S^* = \bigcup_{n=0}^{\infty} S^n$, it is sufficient to show that $S^n \subseteq L$ for all $n \geq 0$. We use (weak) induction on n to show that $x \in S^n \to x \in L$.

I. Let $n = 0$. Then $x \in S^n$ implies $x = \lambda$, which certainly satisfies both (i) and (ii), and therefore $x \in L$.

II. Let $n \geq 0$. Assume that $y \in S^n \to y \in L$. We must show that $x \in S^{n+1} \to x \in L$. Suppose $x \in S^{n+1} = S^n S$. Then either $x = ya$ or $x = yba$ for some $y \in S^n$. The induction hypothesis guarantees that $y \in L$, and therefore y satisfies both (i) and (ii). If $x = ya$ then x does not end in b and does not contain bb since y does not. In this case x satisfies (i) and (ii). If $x = yba$ then certainly x does not end in b. Since y does not end in b and does not contain bb we see that x does not contain bb. In this case also, x satisfies (i) and (ii). In both cases x satisfies (i) and (ii) and therefore $x \in L$. We've shown that $x \in S^{n+1} \to x \in L$.

It follows that $S^n \subseteq L$ for all $n \geq 0$, and therefore $S^* \subseteq L$.

Proof of $L \subseteq S^*$: First observe that properties (i) and (ii) are together equivalent to a single property, namely (iii) every b in x is followed by an a. Indeed a b can in general be followed by only three things: a, b or nothing. (ii) says it cannot be b and (i) says that it cannot be nothing. Thus (i)$\wedge$(ii)$\Leftrightarrow$(iii). Now let $x \in L$, so that x satisfies (iii). We must show that there exists an $n \geq 0$ such that $x \in S^n$, for then $x \in S^*$. Let $k = n_b(x)$. Then by property (iii), x contains k instances of the substring ba, which accounts for $2k$ symbols in x. The remaining $|x| - 2k$ symbols in x must then be instances of the string a. Let $n = n_a(x) = |x| - k = k + (|x| - 2k)$. Then x consists of the concatenation of n strings from S, and hence $x \in S^n$, as required.

b. Let L be the language of all strings $x \in \{a, b\}^*$ that do not contain the substring bb. Then there is no language S such that $S^* = L$.

Proof: Assume, to get a contradiction, that such a language S exists. Observe that both a and b are in L since neither contain the substring bb. By our assumption then, both a and b are in S^*. But this implies that both a and b are also in S. (Observe $\emptyset^* = \{\lambda\}$, $\{a\}^* = \{a^n \mid n \geq 0\}$ and $\{b\}^* = \{b^n \mid n \geq 0\}$, none of which contain both a and b.) Since $\{a, b\} \subseteq S$, we have $\{a, b\}^* \subseteq S^*$ by problem 1.33(a) above. In other words S^* contains *all* strings on a and b, many of which contain the substring bb (the string bb itself for instance.) This contradicts the very definition of $L(= S^*)$ as the set of all strings *not* containing bb. This contradiction shows that no such language S can exist.

4. Problem 1.44abcd (proofs not necessary)

Each case below gives a recursive definition of a subset L of $\{a, b\}^*$. Give a simple non-recursive definition of L in each case.

a. $a \in L$; for any $x \in L$, xa and xb are in L.
b. $a \in L$; for any $x \in L$, bx and xb are in L.
c. $a \in L$; for any $x \in L$, ax and xb are in L.
d. $a \in L$; for any $x \in L$, xb, xa, and bx are in L.

Solution:

a. $L = \{\, x \in \{a, b\}^* \mid x \text{ begins with } a \,\}$
b. $L = \{\, x \in \{a, b\}^* \mid x \text{ contains exactly one } a \,\}$

c. $L = \{ x \in \{a, b\}^* \mid x = a^n b^k$ where $n \geq 1$ and $k \geq 0\}$
d. $L = \{ x \in \{a, b\}^* \mid x$ contains at least one a $\}$

5. Problem 1.52
Prove that for every language $L \subseteq \{a, b\}^*$, if $L^2 \subseteq L$, then $LL^* \subseteq L$.

Proof:
First we prove that $L^2 \subseteq L$ implies $L^n \subseteq L$ for all $n \geq 1$. We proceed by induction on n. The base case ($n = 1$) is easily established since $L \subseteq L$. Now let $n \geq 1$ and assume, for this n, that $L^n \subseteq L$. We must show that $L^{n+1} \subseteq L$. Let $x \in L^{n+1} = LL^n$. Then $x = yz$ for some $y \in L$, $z \in L^n$. The induction hypothesis $L^n \subseteq L$ gives $z \in L$, whence $x \in LL = L^2 \subseteq L$. Therefore $L^{n+1} \subseteq L$ as required.

With this result we now prove $L^2 \subseteq L$ implies $LL^* \subseteq L$. Let $x \in LL^*$. Then $x = yz$ for some $y \in L$ and $z \in L^*$. Therefore $z \in L^n$ for some $n \geq 0$, and hence $z \in L$ by the preceding paragraph. Thus $x \in LL = L^2 \subseteq L$ proving that $LL^* \subseteq L$. The proof is now complete.

6. Problem 1.63
For a string x in the language *Expr* defined in Example 1.19, let $n_a(x)$ denote the number of a's in the string, and let $n_{op}(x)$ stand for the number of operators in x (i.e. the number of occurrences of $+$ or $*$). Show that for every $x \in Expr$, $n_a(x) = 1 + n_{op}(x)$.

Proof: We proceed by structural induction on $x \in Expr$. First recall the recursive definition of $Expr \subseteq \{ a, +, *, (,) \}^*$.

(1) $a \in Expr$
(2) $x, y \in Expr \rightarrow x + y \in Expr$
(3) $x, y \in Expr \rightarrow x * y \in Expr$
(4) $x \in Expr \rightarrow (x) \in Expr$

I. $n_a(a) = 1 = 1 + 0 = 1 + n_{op}(a)$
II. Assume $x, y \in Expr$ satisfy the conditions $n_a(x) = 1 + n_{op}(x)$ and $n_a(y) = 1 + n_{op}(y)$. Then $n_a(x * y) = n_a(x) + n_a(y) = \left(1 + n_{op}(x)\right) + \left(1 + n_{op}(y)\right)$ by the induction hypothesis. Therefore $n_a(x * y) = 1 + \left(n_{op}(x) + 1 + n_{op}(y)\right) = 1 + n_{op}(x * y)$. One shows in a similar manner that $n_a(x + y) = 1 + n_{op}(x + y)$. Finally we have $n_a\big((x)\big) = n_a(x) = 1 + n_{op}(x)$ by the induction hypothesis, so $n_a\big((x)\big) = 1 + n_{op}\big((x)\big)$, as required.

The result $n_a(x) = 1 + n_{op}(x)$ now follows for all $x \in Expr$ by structural induction.

7. Problem 1.65ab
Suppose $L \subseteq \{a, b\}^*$ is defined as follows: $\lambda \in L$; for every $x \in L$, both *xa* and *xba* are in L. Show that for every $x \in L$, both of the statements (a) and (b) below are true.
a. $n_a(x) \geq n_b(x)$
b. x does not contain the substring *bb*.

Solution:
We use structural induction in both parts to show that strings in the recursively defined language L have the specified property.

a. $\forall x \in L:\ n_a(x) \geq n_b(x)$
Proof: First note that $n_a(\lambda) = 0 \geq 0 = n_b(\lambda)$, so the base case is satisfied. Let $x \in L$ and assume $n_a(x) \geq n_b(x)$. We must show that both $n_a(xa) \geq n_b(xa)$ and $n_a(xba) \geq n_b(xba)$ are true. Observe

$$
\begin{aligned}
n_a(xa) &= n_a(x) + 1 \\
&\geq n_b(x) + 1 && \text{by the induction hypothesis} \\
&> n_b(x) \\
&= n_b(xa)
\end{aligned}
$$

Also

$$
\begin{aligned}
n_a(xba) &= n_a(x) + 1 \\
&\geq n_b(x) + 1 && \text{by the induction hypothesis} \\
&= n_b(xba)
\end{aligned}
$$

The result $n_a(x) \geq n_b(x)$ follows for all $x \in L$.

b. $\forall x \in L:$ *x* does not contain *bb*
Proof: We prove the stronger result: $\forall x \in L:$ (i) *x* does not end in *b* and (ii) *x* does not contain *bb*. (Note that this is the same language discussed in problem 3 (1.36ab).) The empty string λ does not end in *b* and does not contain *bb*, so the base case is satisfied. Let $x \in L$ and assume *x* satisfies properties (i) and (ii) above. We must show that both *xa* and *xba* also satisfy (i) and (ii). Obviously neither *xa* nor *xba* end in *b*,. The induction hypothesis says *x* does not contain *bb*, so the same must hold for *xa*. Finally we see *xba* does not contain the substring *bb* since, by the induction hypothesis, *x* does not contain *bb* and does not itself end in *b*. Thus *xa* and *xba* satisfy (i) and (ii), as required.

8. Problem 1.66
Suppose $L \subseteq \{a, b\}^*$ is defined as follows: $\lambda \in L$; for every $x, y \in L$, the strings *axb*, *bxa* and *xy* are also in L. Show that $L = AEqB$, the language of all strings in $\{a, b\}^*$ satisfying $n_a(x) = n_b(x)$.

Solution:
We must show that both $L \subseteq AEqB$ and $AEqB \subseteq L$.
Proof of $L \subseteq AEqB$: We procede by structural induction to show that for all $x \in L$, $n_a(x) = n_b(x)$. First observe that $n_a(\lambda) = 0 = n_b(\lambda)$, which establishes the base case. Assume $x, y \in L$ satisfy $n_a(x) = n_b(x)$ and $n_a(y) = n_b(y)$. We must show that $n_a(axb) = n_b(axb)$, $n_a(bxa) = n_b(bxa)$ and $n_a(xy) = n_b(xy)$. First, observe that

$$
\begin{aligned}
n_a(axb) &= n_a(x) + 1 \\
&= n_b(x) + 1 && \text{by the induction hypothesis} \\
&= n_b(axb)
\end{aligned}
$$

Second, $n_a(bxa) = n_b(bxa)$ follows as above by interchanging *a* and *b*. Finally we have

$$
\begin{aligned}
n_a(xy) &= n_a(x) + n_a(y) \\
&= n_b(x) + n_b(y) && \text{by the induction hypothesis} \\
&= n_b(xy)
\end{aligned}
$$

Therefore $n_a(x) = n_b(x)$ for all $x \in L$, proving $L \subseteq AEqB$.

Proof of $AEqB \subseteq L$: We use (strong) induction on $|x|$ to show that $x \in AEqB \to x \in L$. If $|x| = 0$ then $x = \lambda$, which is in both *AEqB* and *L*, establishing the base case.

Let $|x| > 0$ and assume for any string *y* with $|y| < |x|$, that $y \in AEqB \to y \in L$. We must show that $x \in AEqB \to x \in L$. Suppose $x \in AEqB$. Since $|x| > 0$ we have $|x| \geq 2$. (All strings in *AEqB* must be of even length.) Therefore *x* has one of four forms, depending on its beginning and ending symbols: $x = azb$, $x = bza$, $x = aza$ or $x = bzb$, for some (possibly empty) string *z*. The proof now splits into four cases corresponding to these forms.

<u>Case 1</u>: $x = azb$. Since $x \in AEqB$, we have $n_a(azb) = n_b(azb)$, hence $n_a(z) + 1 = n_b(z) + 1$, therefore $n_a(z) = n_b(z)$ and $z \in AEqB$. Since $|z| < |x|$, the induction hypothesis gives $z \in L$. The definition of L says that $azb \in L$, whence $x \in L$. In this case then, $x \in AEqB \to x \in L$.

<u>Case 2</u>: $x = bza$. This case is similar to the previous one, and we omit it. (Just swap a and b throughout.)

<u>Case 3</u>: $x = aza$. Since $x \in AEqB$, we have $n_a(aza) = n_b(aza)$, so that $n_b(z) = n_a(z) + 2$. Let $k = |z|$. Then *z* has $k + 1$ distinct prefixes of lengths 0 through *k*, respectively. Call them

$$\lambda = p_0,\ p_1,\ p_2,\ \dots\ ,\ p_k = z$$

For each of these prefixes, let $s_i = n_b(p_i) - n_a(p_i)$ denote its number of surplus *b*'s $(0 \leq i \leq k)$. The integer sequence s_i starts at $s_0 = 0$ and ends at $s_k = 2$. It also steps by $+1$ or -1 from term to term. Therefore there must exist an index *j* in the range $0 \leq j \leq k$ for which $s_j = 1$. The prefix p_j satisfies $n_b(p_j) = n_a(p_j) + 1$. Since $n_b(z) = n_a(z) + 2$, the suffix consisting of all letters in *z* other than those in p_j also has one surplus b. We've shown that *z* factors as $z = uv$ where both $n_b(u) = n_a(u) + 1$ and $n_b(u) = n_a(u) + 1$. Therefore $x = (au)(va)$ where $n_a(au) = n_b(au)$ and $n_a(va) = n_b(va)$. Thus both *au* and *va* belong to *AEqB*, and since $|au| < |x|$ and $|va| < |x|$, the induction hypothesis guarantees *au* and *va* belong to *L*. Finally, the definition of L says $x = (au)(va) \in L$. Therefore in this case also, $x \in AEqB \to x \in L$.

<u>Case 4</u>: $x = bzb$. Again this case is entirely similar to the previous (by swapping *a* and *b*) and is left to the reader.

In all cases $x \in AEqB \to x \in L$, so we have $AEqB \subseteq L$, as claimed.