CS 341 Homework 3
Languages and Regular Expressions

1. Describe in English, as briefly as possible, each of the following (in other words, describe the language defined by each regular expression):
(a) L(((a*a) b) ∪ b **)**
(b) L((((a*b*)*ab) ∪ ((a*b*)*ba))(b ∪ a)* **)**

2. Rewrite each of these regular expressions as a simpler expression representing the same set.
(a) ∅* ∪ a* ∪ b* ∪ (a ∪ b)*
(b) ((a*b*)* (b*a*)*)*
(c) (a*b)* ∪ (b*a)*

3. Let Σ = {a, b}. Write regular expressions for the following sets:
(a) All strings in Σ* whose number of a's is divisible by three.
(b) All strings in Σ* with no more than three a's.
(c) All strings in Σ* with exactly one occurrence of the substring aaa.

4. Which of the following are true? Prove your answer.
(a) baa ∈ a*b*a*b*
(b) b*a* ∩ a*b* = a* ∪ b*
(c) a*b* ∩ c*d* = ∅
(d) abcd ∈ (a(cd)*b)*

5. Show that L((a ∪ b)*) = L(a* (ba*)*).

6. Consider the following:

(a) ((a ∪ b) ∪ (ab))*
(b) (a^{+} $a^{n}b^{n}$)
(c) ((ab)* ∅)
(d) (((ab) ∪ c)* ∩ (b ∪ c*))
(e) (∅* ∪ (bb*))

(i) Which of the above are “pure” regular expressions?
(ii) For each of the above that is a regular expression, give a simplified equivalent “pure” regular expression.
(iii) Which of the above represent regular languages?

7. True - False: For all languages L1, L2, and L3
(a) (L1L2)* = L1*L2*
(b) (L1 ∪ L2)* = L1* ∪ L2*
(c) (L1 ∪ L2) L3 = L1 L3 ∪ L2 L3
(d) (L1 L2) ∪ L3 = (L1 ∪ L3) (L2 ∪ L3)
(e) $L1^{+}$)* = L1*
(f) $(L1^{+})^{+} = L1^{+}$
(g) $(L1*)^{+}$ = (L1+)*
(h) L1* = $L1^{+}$ ∪ ∅
(i) (ab)*a = a(ba)*
(j) (a ∪ b)* b (a ∪ b)* = a* b (a ∪ b)*
(k) [(a ∪ b)* b (a ∪ b)* ∪ (a ∪ b)* a (a ∪ b)*] = (a ∪ b)*
(l) [(a ∪ b)* b (a ∪ b)* ∪ (a ∪ b)* a (a ∪ b)*] = $(a ∪ b)^{+}$
(m) [(a ∪ b)* b a (a ∪ b)* ∪ a*b*] = (a ∪ b)*

(n) (L1L2L3)* = L1*L2*L3*
(o) (L1* ∪ L3*) = (L1* ∪ L3*)*
(p) L1* L1 = L1+
(q) (L1 ∪ L2)* = (L2 ∪ L1)*
(r) L1* $(L2 \cup L3)^+$ = (L1* $L2^+$ ∪ L1* $L3^+$)
(s) ∅ L1* = ∅
(t) ∅ L1* = {ε}
(u) (L1 - L2) = (L2 - L1)
(v) ((L1 L2) ∪ (L1 L3))* = (L1 (L2 ∪ L3))*

8. Let L = {w ∈ {a, b}* : w contains bba as a substring}. Find a regular expression for {a, b}* - L.

9. Let Σ = {a,b}. For each of the following sets of strings (i.e., languages) L, first indicate which of the example strings are in the language and which are not. Then, if you can, write a concise description, in English, of the strings that are in the language.
Example strings: (1) aaabbb, (2) abab, (3) abba, (4) ε

(a) L = {w : for some u ∈ Σ*, w = u^Ru}

(b) L = {w : ww = www}

(c) L = {w : for some u ∈ Σ*, www = uu}

10. Write a regular expression for the language consisting of all odd integers *without* leading zeros.

11. Let Σ = {a, b}. Let L = {ε, a, b}. Let R be a relation defined on Σ* as follows: ∀xy, xRy iff y = xb. Let R′ be the reflexive, transitive closure of L under R. Let L′ = {x : ∃y ∈ L such that yR′x}. Write a regular expression for L′.

Solutions

1. (a) Any string of a's and/or b's with zero or more a's followed by a single b.
(b) Any string of a's and/or b's with at least one occurrence of ab or ba.

2. (a) ∅* = {ε}, and ε ⊆ (a ∪ b)*.
a* ⊆ (a ∪ b)*.
b* ⊆ (a ∪ b)*. So since the first three terms describe subsets of the last one, unioning them into the last one doesn't add any elements. Thus we can write simply (a ∪ b)*.

(b) To solve this one, we'll use some identities for regular expressions. We don't have time for an extensive study of such identities, but these are useful ones:

((a*b*)* (b*a*)*)* =
Using (A*B*)* = (A ∪ B)* (Both simply describe any string that is composed of elements of A and elements of B concatenated together in any order)
((a ∪ b)*(b ∪ a)*)* =
Using (A ∪ B) = (B ∪ A) (Set union is commutative)
((a ∪ b)*(a ∪ b)*)* =
Using A*A* = A*
((a ∪ b)*)* =
Using (A*)* = A*
(a ∪ b)*

(c) $(a^*b)^* \cup (b^*a)^* = (a \cup b)^*$ (In other words, all strings over {a, b}.) How do we know that? $(a^*b)^*$ is the union of ε and all strings that end in b. $(b^*a)^*$ is the union of ε and all strings that end in a. Clearly any string over {a, b} must either be empty or it must end in a or b. So we've got them all.

3. (a) The a's must come in groups of three, but of course there can be arbitrary numbers of b's everywhere. So:

$(b^*ab^*ab^*a)^*b^*$

Since the first expression has * around it, it can occur 0 or more times, to give us any number of a's that is divisible by 3.

(b) Another way to think of this is that there are three optional a's and all the b's you want. That gives us:

$b^* (a \cup \varepsilon)\ b^* (a \cup \varepsilon)\ b^* (a \cup \varepsilon)\ b^*$

(c) Another way to think of this is that we need one instance of aaa. All other instances of aa must occur with either b or end of string on both sides. The aaa can occur anywhere so we'll plunk it down, then list the options for everything else twice, once on each side of it:

$(ab \cup aab \cup b)^* \quad aaa \quad (ba \cup baa \cup b)^*$

4. (a) True. Consider the defining regular expression: $a^*b^*a^*b^*$. To get baa, take no a's, then one b, then two a's then no b's.

(b) True. We can prove that two sets X and Y are equal by showing that any string in X must also be in Y and vice versa. First we show that any string in $b^*a^* \cap a^*b^*$ (which we'll call X) must also be in $a^* \cup b^*$ (which we'll call Y). Any string in X must have two properties: (from b^*a^*): all b's come before all a's; and (from a^*b^*): all a's come before all b's. The only way to have both of these properties simultaneously is to be composed of only a's or only b's. That's exactly what it takes to be in Y.

Next we must show that every string in Y is in X. Every string in Y is either of the form a^* or b^*. All strings of the form a^* are in X since we simply take b^* to be b^0, which gives us $a^* \cap a^* = a^*$. Similarly for all strings of the form b^*, where we take a^* to be a^0.

(c) False. Remember that to show that any statements is false it is sufficient to find a single counterexample:

$\varepsilon \in a^*b^*$ and $\varepsilon \in c^*d^*$. Thus $\varepsilon \in a^*b^* \cap c^*d^*$, which is therefore not equal to ∅.

(d) False. There is no way to generate abcd from $(a(cd)^*b)^*$. Let's call the language generated by $(a(cd)^*b)^*$ L. Notice that every string in L has the property that every instance of $(cd)^*$ is immediately preceded by a. abcd does not possess that property.

5. That the language on the right is included in the language on the left is immediately apparent since every string in the right-hand language is a string of a's and b's. To show that any string of a's and b's is contained in the language on the right, we note that any such string begins with zero or more a's. If there are no b's, then the string is contained in a^*. If there is at least one b, we strip off any initial a's as a part of a^* and examine the remainder. If there are no more b's, the remainder is in ba^*. If there is at least one more b to the right, then we strip of the initial b and any following consecutive a's (a string in ba^*) and examine the remainder. Repeat the last two steps until the end of the string is reached. Thus, every string of a's and b's is included in the language on the right.

6. (i) a, c, e (b contains superscript n; d contains ∩)

(ii) (a) = $(a \cup b)^*$

(c) = ∅

(e) = b^*

(iii) a, c, d, e (b is $\{a^m b^n : m > n\}$, which is not regular)

7. (a) F, (b) F, (c) T, (d) F, (e) T, (f) T, (g) T, (h) F, (i) T, (j) T, (k) F, (l) T, (m) T, (n) F, (o) F, (p) T (by def. of +), (q) T, (r) F, (s) T, (t) F, (u) F, (v) T.

8. $(a \cup ba)^* (\varepsilon \cup b \cup bbb^*) = (a \cup ba)^*b^*$

9. (a) (1) no (2) no, (3) yes, (4) yes
L is composed of strings whose second half is the reverse of the first half.

(b) (1) no (2) no (3) no (4) yes
L contains only the empty string.

(c) (1) no (2) yes (3) no (4) yes
L contains strings of even length whose first half is the same as the second half. To see why this is so, notice that $|uu|$ is necessarily even, since it's $|u|$ times 2. So we must assure that $|www|$ is also even. This will only happen if $|w|$ is even. To discover what *u* is for any proposed *w*, we must first write out *www*. Then we split it in half and call that *u*. Suppose that *w* can be described as the concatenation of two strings of equal length, *r* and *s*. (We know we can do this, since we already determined that $|w|$ is even.) Then *w* will be equal to *rs* and *www* will be *rsrsrs*. So *u* must equal both *rsr* and *srs*. There can only be such a *u* if *r* and *s* are the same.

10. $(\varepsilon \cup ((1\text{-}9)(0\text{-}9)^*))(1\cup3\cup5\cup7\cup9)$, or, without using ε or the dash notation,

$$(1\cup3\cup5\cup7\cup9) \cup ((1\cup2\cup3\cup4\cup5\cup6\cup7\cup8\cup9)\ (0\cup1\cup2\cup3\cup4\cup5\cup6\cup7\cup8\cup9)^*\ (1\cup3\cup5\cup7\cup9))$$

11. Whew. A lot of formalism. The key is to walk through it one step at a time. It's good practice. R relates pairs of strings that are identical except that the second one has one extra b concatenated on the end. So it includes, for example, {(a, ab), (ab, abb), (abb, abbb), (b, bb), (bb, bbb), …}. Now we have to compute R′. Consider the element a. First, we must add (a, a) to make R′ reflexive. Now we must consider transitivity. R gives us (a, ab). But it also gives us (ab, abb), so, by transitivity, R′ must contain (a, abb). In fact, a must be related to all strings in the language ab*. Similarly ε must be related to all strings in εb* or simply b*. And b must be related to all strings in bb*. We could also notice many other things, such as the fact that ab is related to all strings in abb*, but we don't need to bother to do that to solve this problem. What we need to do is to figure out what L′ is. It's all strings that are related via R′ to some element of L. There are three elements of L, {ε, a, b}. So L′ = b* ∪ ab* ∪ bb*. But every string in bb* is also in b*, so we can simplify to b* ∪ ab*.