CS 360 Fall 2013
Assignment 1 - Solutions

1. (a) False. Consider $Y = \{(-\inf, 0), (0, \inf)\}$. Each set in Y is infinite, yet their intersection $\bigcap Y$ is empty, in particular finite.

   (b) False. Consider $Y = \{\{1,2\}\}$ and $Z = \{\{1,2\},\{1\}\}$. As each set in Y is also in Z, $Y \subseteq Z$. However, $\bigcap Y = \{1,2\}$ and $\bigcap Z = \{1\}$, so $\bigcap Y \not\subseteq \bigcap Z$. In general if we take the intersection of more sets, we will get a smaller (or equal) set.

   Of course there are other acceptable counter-examples as well.

2. Claim: $\bigcap Y = \emptyset$.

   Proof by contradiction: assume that the claim is not true and there exists $x \in \bigcap Y$. Now consider the set $B = \mathbf{N} \setminus \{x\}$. We know that $B \in Y$ because $\mathbf{N} \setminus B$ is a subset of $\mathbf{N}$ and has a finite number of elements (i.e., one element). Furthermore, $x \notin B$ and therefore, $x \notin B \bigcap Y = \bigcap Y$ which is a contradiction.

3. To prove the equivalence of the two statements, we prove implications in both directions.

   (a) $\epsilon \in L \Rightarrow L \subseteq LL$: Take arbitrary $w \in L$. By assumption $\epsilon \in L$, so $w = w\epsilon \in LL$.

   (b) $\epsilon \in L \Leftarrow L \subseteq LL$: Let $|w|$ denote the length of a string w. Note that any length must be non-negative. There is an element of minimal length in L by the well-ordering principle; call this w. By assumption ($L \subseteq LL$) w is also in LL, that is there exists $x, y \in L$ such that $w = xy$. Then we must have $|w| = |x| + |y|$. But by the choice of w: $|x|, |y| \geq |w|$. Combining these two facts:

   $$|w| = |x| + |y| \geq 2|w| \;\Rightarrow\; 0 \geq |w|$$

   . Hence $|w|$ must be 0, and the only string of length zero is ϵ.

4. Let B be the set of all finite strings of the form $a^iba^jca^k$ where $k = i * j$.

   Claim: $I(A, P) \subseteq B$.

   Proof by structural induction. Base case: $abaca$ is clearly in B where $i = j = k = 1$. Inductive step: let $s = a^iba^jca^{i*j}$ be a member of $I(A, P)$. Then

   $F_1(s) = a^{i+1}ba^jca^{i*j+j} = a^{i+1}ba^jca^{(i+1)*j} \in B$

   $F_2(s) = a^iba^{j+1}ca^{i*j+i} = a^{i+1}ba^jca^{i*(j+1)} \in B$

   Therefore, $L_{times} = I(A, P) \subseteq B$ and every member of L_{times} is in the form of $a^iba^jca^{i*j}$. Now we can answer to the questions:

   - (i) $ababab$ is clearly not in the form of $a^iba^jca^{i*j}$ (e.g., it doesn't have a "c") and therefore, $ababab \notin L_{times}$.

- (ii) $aaabaacaaaaaa \in L_{times}$ because we can produce it with the following sequence of operations

  $F_1(F_1(F_2(abaca))) = F_1(F_1(abaacaa)) = F_1(aabaacaaaa) = aaabaacaaaaaa$
- (iii) $aabacaaaa$ is not of the form $a^iba^jca^{i*j}$ $(2 * 1 \neq 4)$ and therefore, $ababab \notin L_{times}$.

5. We use structural induction to prove the claim that the wine glass (cup 1) will always contain more wine than water. Note that we may pour any amount of liquid from one cup to the other as long as it does not overflow. We consider one cup to hold one unit of volume. A key observation is that the total amount of water (and wine) in the two cups is always 0.75, that is

$$a_1 + a_2 = \frac{3}{4} = i_1 + i_2,$$

where a_j denotes the amount of water in cup j, and similarly i_j denotes the amount of wine in cup j. In particular:

$$a_1 > i_1 \Leftrightarrow a_2 < i_2, \tag{*}$$

That is, cup 1 holds more wine exactly when cup 2 holds more water. Indeed, if it wasn't so then either $a_1 \leq i_1$ or $a_2 \geq i_2$. Then $a_1 + a_2 < i_1 + i_2$ from term-by-term inequalities, which is a contradiction to our key observation.

Let X be the set of all states attainable by pouring liquid from cup to cup. Our core set is $A = \{$ cup 1 is $\frac{3}{4}$ full with wine, cup 2 is $\frac{3}{4}$ full with water$\}$. The set of operations is $P = \{P_1 :$ pour from cup 1 to cup 2, $P_2 :$ pour from cup 2 to cup 1$\}$. Finally, define $B = \{$ cup 1 has more wine than water $\}$. To prove the claim we show $I(A, P) \subseteq B$:

Induction base: $A \subseteq B$, as cup 1 has $\frac{3}{4}$ wine and no water.

Induction step: Fix any $S \in B$; by definition $\frac{\text{wine}}{\text{water}} > \frac{1}{2}$ in cup 1. Now apply each operation to S and check if the resulting states are still in B.

- $P_1(S) \in B$: Pouring well mixed liquid out of this cup does not change the ratio of water to wine. ✓
- $P_2(S) \in B$: By (*), $\frac{\text{wine}}{\text{water}} < \frac{1}{2}$ in cup 2. Pouring out of this cup does not change this ratio, and so it also holds for $P_2(S)$. But by using (*) again, we get $\frac{\text{wine}}{\text{water}} > \frac{1}{2}$ in cup 1. ✓

This shows us that the claim indeed holds. Note that there are other approaches to prove this claim, for example it is possible to write down an algebraic expression for what amounts of wine and water each cup will hold after each pouring, and showing by the manipulation of inequalities that the amount of wine is more than the amount of water in cup 1.

6. $L(r_1) = L(\lambda + (0+1)^*1) = \{\lambda\} \cup (\{0,1\}^*.\{1\}) = \{\lambda$ or any member of $\{0,1\}^*$ that ends with $1\}$.

   $L(r_2) = L((0^*1)^*) = (\{0\}^*.\{1\})^*$

   We need to prove that $L(r_1) \subseteq L(r_2)$ and $L(r_2) \subseteq L(r_1)$.

   Claim: $L(r_1) \subseteq L(r_2)$. Proof: Let x be an arbitrary member of $L(r_1)$. There are two cases.

   (a) $x = \lambda$. However, $\lambda \in L(r_2)$ and therefore $x \in L(r_2)$.

   (b) x is a sequence of 1s and 0s ending with 1. Without loss of generality, assume that x contains k letters of 1 in it. Now, we can split x to k substrings like this: let $index(j,x)$ be the index of the j-th "1" in x (and $index(0,x) = 0$). Then j-th substrings is defined as the sequence of letters starting from $index(j-1,x)$ and ending with $index(j,x)$. So each of these k substrings is a member of $L(0^*1)$ and the concatenation of k strings will be a member of $L((0^*1)^*)$. Therefore, $x \in L(r_2)$

   Claim: $L(r_2) \subseteq L(r_1)$. Proof: Let x be an arbitrary member of $L(r_2)$. If $x = \lambda$ then we know that $x \in L(r_1)$. Otherwise, x is a concatenation of $k > 0$ strings of the form 0^*1 and therefore, x has a 1 at the end. However, $L(r_1)$ contains all of such strings, and we can conclude that $x \in L(r_1)$.

7. (a) $r = ((\lambda + a + aa)b)^*aa(b(\lambda + a + aa))^*$. The aa in the middle ensures that there is at least one aa in matching strings. By the use of the b delimiters between the (at most 2 long) sequences of as we see that there will be no sequences of as longer than 2. This shows $L(r) \subseteq L$ where L denotes the specified language. Now given any $w \in L$, split it just before (prefix) and after (suffix) the first occurrence of aa. The aa will be matched by the aa in the expression, the prefix (which must be empty or end with b) by $((\lambda + a + aa)b)^*$ and the suffix (must be empty or start with b) by $(b(\lambda + a + aa))^*$. Indeed for any word w having no more than 2 consecutive as and ending with a b the first pattern will match it: tokenize w by splitting after each b and one of the three choices in the expression will have to match each token. The justification for the suffix is similar.

   (b) $r = a(a+b)^*b + b(a+b)^*a$. Note that $(a+b)^*$ matches any string. Since there are two letters in the alphabet, the only possibilities for having the last and first letters different is starting with a and ending with b (with any sequence of letters inbetween), or vice versa. These correspond to the 2 main building blocks of r, which joined by 'or' (+).