CS 381 – Prelim2 Solutions

1. Let $L_1 = \{a^i b^{2^k} \mid i > 0, k \geq 0\}$. Prove that $L_2 = \{b^i \mid i \geq 0\} \cup L_1$ does satisfy the pumping lemma for regular languages ($25 Points$).

Solution 1:

We must find a number n such that every $w \in L_2$ with $|w| > n$ can be pumped: there exists x, y, z such that $xyz = w$, $|xy| \leq n$, $|y| \geq 1$, and for any $i \in \mathbb{N}$ it is true that $xy^iz \in L_2$. Lets say $n = 1$. Consider all words $w \in L_2$ of length 2 or greater.

Case 1: $w = b^i$. We know $i \geq 1$ since choosing $n = 1$ means $|w| > 1$. Because $|xy| \leq 1$ and $|y| \geq 1$ it must be that $|x| = 0$ and $|y| = 1$. In other words $x = \epsilon$, $y = b$ and z is thus b^{i-1}. Clearly, $|xy| = |\epsilon b| \leq 1$ and $|y| = |b| \geq 1$. Pumping w to $w' = xy^jz = b^{i+j-1}$ for any non-negative j, its clear that $w' \in L_2$ because it is of the form b^{i+j-1} and $i + j - 1 \geq 0$ as $j \geq 0$ and $i \geq 1$.

Case 2: $w = a^i b^{2^k}$. We know $i > 0$ by definition. Once again, we have to pick $|x| = 0$ and $|y| = 1$, which means $x = \epsilon$, $y = a$ because the first character of w is a and z is the rest of the string. Clearly, $|xy| = |\epsilon a| \leq 1$ and $|y| = |a| \geq 1$. Pumping w to $w' = xy^jz = a^{i+j-1}b^{2^k}$ for any non-negative j, its clear that $w' \in L_2$ because $i + j - 1$ is either 0 or greater than 0. If it is 0 then we are of the form $b^{i=2^k}$ and $2^k \geq 0$ so we are in L_2 (Note: $L_2 = \{b^i \mid i \geq 0\} \cup L_1$). If it is greater than 0, then we are still of the form $a^{i=i+j-1}b^{2^k}$ and $i > 0, k \geq 0$ so we are in L_1 and thus in L_2.

Thus, L_2 satisfies the pumping lemma because for any $w \in L$ with $|w| > 1$, we can partition w into substrings xyz with $|xy| \leq n$ and $|y| \geq 1$ so that w is pumpable: $xy^iz \in L_2$ for all $i \in \mathbb{N}$.

Comments 1: Grading of this problem proceeded as follows: 5 points for choosing a number n; n had to be a number. Not defining n, or defining it as a variable were common mistakes. 5 points for defining x, y, z such that for all w, $|w| > n$ (the n you chose), $xyz = w$. 5 points for showing that the properties of the Pumping Lemma for regular languages hold for your definitions of x, y, z and your choice of n (ie $|xy| \leq n$ and $|y| \geq n$) and 10 points for showing that for all $j \in \mathbb{N}$, $xy^jz \in L_2$. Since there were two cases to be considered, the point values to these steps were split between the two cases. Case 2 had the added difficulty of the $j = 0$ case which was worth 3 of its 5 *show* points.

2. Prove that L_2 described above is not a regular language. (10 bonus points)

Solution 2:

We do this by constructing an infinite set of words $S = \{w_1, w_2, \ldots\}$, all of which are pairwise non-equivalent under the relation R_L (ie: for $i \neq j$, we can find some string z so that $w_iz \in L_2$ while $w_jz \notin L_2$ or vice versa).

Define $w_i = ab^{2^i}$ and $S = \{w_1, w_2, \ldots\}$. Take any two distinct w_i and w_j. Without loss of generality, assume $i < j$. Now define $z = b^{2^i}$. Observe that $w_i z = ab^{2^i} b^{2^i} = ab^{2^{i+1}} \in L_2$. However, $w_j z = ab^{2^j} b^{2^i} \neq ab^{2^k}$ for any integer k because for $i < j$ we know that $2^j < 2^j + 2^i < 2^j + 2^j = 2^{j+1}$. Thus $w_j z \notin L_2$.

Thus, we can conclude that R_L partiitions L_2 into infinitely many equivalence classes (i.e., R_L has infinite rank), so by the Myhill-Nerode theorem, L_2 is not a regular language.

Grading:

The most common correct solution submitted was along the lines of the proof given above. 3 points were awarded for giving an infinite set of equivalence classes, 5 points for proving it, and 2 points for knowing the name of the theorem.

It was also possible to prove this using Kozen's version of the pumping lemma, which is stronger than the one we saw in class–a few people did this.

Another way to prove this is to use the fact that regular languages are closed under complement and intersection, combined with the pumping lemma, as follows: Let $L_3 = \{b^i | i \geq 0\}$, and show (easily) that L_3 is regular. So, assuming L_2 is regular, then $L_1 = L_2 \cap \overline{L_3}$ is regular. However, one can use the pumping lemma to show that L_1 is not regular, so our assumption that L_2 was regular must be incorrect.

Common Errors:

A disturbing number of people tried to use the pumping lemma (the one from class) to prove this, after showing in Problem 1 that L_2 *satisfies* the conditions of the pumping lemma.

Others incorrectly applied the closure properties of regular languages. For example, it is not the case that non-regular languages are closed under union with regular languages; e.g., $L \cup \Sigma^*$ is regular for any language L, even if it is not regular. It is also not true that AB is non-regular for any non-regular language B and any language A; consider $A = \emptyset$.

3. Let $L = \{w \in \{a, b, c\}^* \mid \#_a(w) + \#_b(w) > \#_c(w)\}$. Prove that $\forall i, j, k, l \in \mathbb{N}, a^i b^j \sim_{R_L} a^k b^l$ if and only if $i + j = l + k$. (25 points)

Solution 3:

If $i + j \neq l + k$ then $x = a^i b^j \nsim_{R_L} y = a^k b^l$ because if we pick $z = c^{max(i+j-1, k+l-1)}$ then only one of xz and yz is in L. Whichever has a smaller index sum is clearly not in L because there are too many cs. Whichever has the larger index sum is fine, however, because it has the maximum allowed c's.

If $i + j = l + k$ then $x = a^i b^j \sim_{R_L} y = a^k b^l$. Lets consider any string z that we could append to x or y. Define $n_{a,b,c}$ as $n_a = \#_a(z)$, $n_b = \#_b(z)$, and $n_c = \#_c(z)$. Now consider whether or not xz and yz are in L. Well, $\#_a(xz) + \#_b(xz) = i + j + n_a + n_b$ and $\#_c(xz)\} = n_c$. Also, $\#_a(yz) + \#_b(yz) = k + l + n_a + n_b$ and $\#_c(yz)\} = n_c$. Because $i + j = k + l$ these expressions are the same: $\#_a(xz) + \#_b(xz) = \#_a(yz) + \#_b(yz)$ and $\#_c(xz) = \#_c(yz)$. Therefore, if either $xz \in L$ or $yz \in L$ then the other string is also in L.

We have shown both directions of the if and only if, and thus can conclude that $a^i b^j \sim_{R_L} a^k b^l$ iff $i + j = l + k$.

A common error:
Recall the definition for the relation R_L is:
x R_L y iff: for all strings $z \in \Sigma^*$, xz $\in$ L $\Leftrightarrow$ yz $\in$ L; *not* just: x$\in$L $\Leftrightarrow$ y$\in$L.

Grading policy:
i)Remember to prove both directions, each direction 12-13 points.
ii)When proving $i+j = l+k \Rightarrow xR_Ly$, the string z you append to x or y can be any string in $\{a, b, c\}^*$, not only c^n. If the z you chose wass not general enough, you lost lose 5-7 points.
iii)If you misunderstood the meaning of R_L, you lost more than 10 points. it depends on your performance on the rest work.
iv)You lost 2-5 points if you didn't make your idea clear.

4. Let $C_1 = \{a^i b^{2i} c^k \mid i, k \in \mathbb{N}\}$ and $C_2 = \{a^n b^l c^l \mid n, l \in \mathbb{N}\}$.

Problem 4i: Describe a context free grammar G such that $L(G) = C_1$. (25 points)

Solution 4i:
Define $G = (\{S, X_{abb}, C\}, \{a, b, c\}, P, S)$ where our production rules P are given as follows. $S \to XC$. $X_{abb} \to aXbb \mid \epsilon$. $C \to cC \mid \epsilon$. For this grammar, $L(G) = C_1$.

Problem 4ii: Prove that $C_1 \cap C_2$ is not a CFL. (25 points)

Problem 4ii: Prove that $C_1 \cap C_2$ is not a CFL.

Solution 4ii:
Words $w \in C_1 \cap C_2$ are of the form $a^i b^{2i} c^k$ and also $a^n b^l c^l$. This means that $k = 2i$. So our language $C = \{a^i b^{2i} c^{2i} \mid i \in \mathbb{N}\}$. We can prove that C is not a CFL using the pumping lemma for CFLs. Lets play the CFL demon game!

The demon gives us n. We must reply with some word longer than n that cannot be pumped. Lets pick $w = a^n b^{2n} c^{2n}$ because clearly $|w| > n$.

The demon partitions w into $x_1 x_2 x_3 x_4 x_5$ with $|x_2 x_3 x_4| \leq n$ and $|x_2 x_4| \geq 1$. Lets switch on the demon's $w' = x_2 x_3 x_4$. If w' contains only one symbol type (either a, b, or c) then clearly we cannot pump w and stay in C because we are not allowed to change the ratio of the number of different symbols. Next, w' cannot contain a and b and c symbols because b appears $2n$ times between the last a and the first b and $|w'| \leq n$. Finally, if w' contains any two symbols (either $a + b$, $b + c$, or $a + c$) then w is still unpumpable. Pumping such a string would change the ratio of the symbols in w' to the symbols not in w', and for a word to be in C it must have exactly the right ratio between all symbols.

Thus, regardless of the demon's choice of n we can pick a word w so that regardless of the demon's partitioning of w the word is not pumpable. The CFL pumping lemma does not hold for C, and therefore C is not a CFL.