CS/ECE 374 Spring 2023 William Cheng (shihuac2@illinois.edu)

Homework 4 Problem 1

Solution: (a) i. Let $F = \{01^k | k > 0\}$. Let x, y be arbitrary strings in F. Then $x = 01^i$ and $y = 01^j$ for some positive integers $i \neq j$. Let $z = 2^i$. Then $xz = 01^i2^i \in L$ by definition of L. However, $yz = 01^j2^i \notin L$ because $1 + j \neq 1 + i$. Therefore, F is a fooling set for L. Since F is infinite, L is not regular.

ii. Let F be the language described by 0^+1. Let x, y be arbitrary strings in F. Then $x = 0^i1$ and $y = 0^j1$ for some positive integers $i \neq j$. Let $z = 0^i$. Then $xz = 0^i10^i \notin L$ because the two blocks of 0's have equal lengths. But $yz = 0^j10^i \in L$ since $i \neq j$. Therefore, F is a fooling set for L. Since F is infinite, L is not regular.

iii. Let $f(h) = \lceil h \log_2 h \rceil$. Then

$$
\begin{aligned}
f(h+1) - f(h) &= \lceil (h+1)\log_2(h+1) \rceil - \lceil h \log_2 h \rceil \\
&\geq \lceil (h+1)\log_2(h+1) \rceil - \lceil h \log_2 (h+1) \rceil \\
&\geq (h+1)\log_2(h+1) - h\log_2(h+1) - 1 \\
&\geq \log_2(h+1) - 1 \\
&> \log_2(h+1) - 2 \\
&> \log_2(\frac{h+1}{4})
\end{aligned}
$$

Let n be an arbitrary positive integer. Let $h = 4 * 2^n - 1$. Then $n = \log_2(\frac{h+1}{4})$. Therefore $f(h+1) - f(h) > n$. Consider the set $F_n = \{0^{f(h)}, 0^{f(h)+1}, ..., 0^{f(h+1)-1}\}$. It has at least n elements.

Consider arbitrary strings $x, y \in F_n$, $x = 0^{f(h)+i}$ and $y = 0^{f(h)+j}$, such that i, j are non-negative integers and $i < j$. Let $z = 0^{f(h+1)-f(h)-j}$. Then

- $yz = 0^{f(h+1)}$. $yz \in L$ by definition.
- $xz = 0^{f(h+1)+i-j}$. $f(h+1) + i - j > f(h)$ since $j < f(h+1) - f(h)$ by definition of F_n. $f(h+1) + i - j < f(h+1)$ since $i < j$. Therefore $xz \notin L$ because the number of zeros is not a possible value of f.

Therefore, F_n is a fooling set of length $\geq n$ for L. Since F_n exists for any $n > 0$, L is not regular.

(b) Let $F_k = \{w \in \{0,1\}^* \mid |w| = k\}$. Let x, y be arbitrary strings in F_k. To find the distinguishing suffix z, let f be a function such that $z = f(x, y)$. f is defined recursively as follows:

- If $\#(0, x) \neq \#(0, y)$, then $z = 0^{\#(1,x)}1^{\#(0,x)}$.
- If $\#(0, x) = \#(0, y)$, and $x = ax', y = ay'$ where $a \in \{0, 1\}$, then $z = (f(x', y') \cdot 01)$.

Explanation. F_k is a set containing all bitstrings of length k. Intuitively, the function finds from left to right the first position where x and y's suffixes, x' and y', have different number of 0's (thus different number of 1's since their length is identical). Then, a suffix z' is added to x' such that $x'z'$ has equal number of 0's and 1's. It is easily known that $y'z'$ does not have equal number of 0's and 1's because the number of 0's and 1's in y' is different from x'.

Finally, a block of 01's is added to z' to make z, until $|x'z| = |y'z| = 2k$. The addition of the 01 block will still make $x'z$ have equal number of 0's and 1's, and make $y'z$ have unequal number of 0's and 1's. Therefore, by definition, $xz \notin L_k$ and $yz \in L_k$. Thus, F_k is a fooling set for L_k. $|F_k| = 2^k$.

(c) L' is regular because it is finite and can be represented with the regular expression $\sum w \in L'$. Then $L' \cap L$ is also regular since it is a subset of L' and thus finite. Assume $L \setminus L'$ is regular. Then $(L \setminus L') \cup (L \cap L') = L$ must be regular by closure of union operation in regular languages. This is a contradiction because L is not regular. Therefore, $L \setminus L'$ must be non-regular.

Example: Let $L = \{0^k 1^k | k \geq 0\}$. Let $L' = L(0^* 1^*)$. Then L is non-regular and L' is regular. $L \setminus L' = \emptyset$ which is a regular language.

■

CS/ECE 374 Spring 2023	William Cheng (shihuac2@illinois.edu)
Homework 4 Problem 2	

Solution: (a)

$$S \to \epsilon \mid aSc \mid aSd \mid bSc \mid bSd$$

Explanation. ϵ is clearly in this language. For any S in this language, we remove one letter from $\{a, b\}$ and one letter from $\{c, d\}$, and the resulting string is still in this language since $i + j = k + l \implies i + j - 1 = k + l - 1$. Therefore, we can define S recursively as shown above.

(b)

$$S \to \epsilon \mid 0S222 \mid 1S222$$

Explanation. ϵ is clearly in this language. For any S in this language, we remove three 2's at the end, and remove one 0 or 1 in the front, the resulting string is still in the language since $3(i + j) - 3 = 3(i + j - 1)$. Therefore, we can define S recursively as shown above.

(c)

$S \to P \mid B\#S \mid S\#B \mid R$	
$B \to \epsilon \mid 1B \mid 0B$	bitstrings
$P \to \epsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$	palindromes
$R \to \#C\# \mid 0R0 \mid 1R1$	strings in L with $x_1 = x_k^R$
$C \to B \mid C\#B$	bitstrings separated by $\#$'s

Explanation. Consider the two cases for a string $x_1\#x_2\#...\#x_k$ to be in L.

i. There exists some $i = j$, such that $x_i = x_j^R$. In other words, x_i is a palindrome. From the start symbol S, all these strings can be described with $S \to P \mid B\#S \mid S\#B$, where P represents palindromes and B represents bitstrings.

ii. There exists some $i \neq j$, such that $x_i = x_j^R$.
First consider the subcase where x_1 and x_k, the first and last substrings separated by $\#$, are reversals of each other. In other words, $(i, j) = (1, k)$. All these strings are described by the non-terminal R.
Then, all strings in case (ii) can be described by $S \to R \mid B\#S \mid S\#B$.

Then consider an arbitrary string in L with $x_i = x_j^R$ for some i, j. Either $i = j$ or $i \neq j$, so it would fall into exactly one case above. Therefore, the union of these cases describes L, which can be represented using the CFG $S \to P \mid B\#S \mid S\#B \mid R$.

(d)

$S \to M \mid L$	$\{1^m 0^n \mid m \neq n\}$
$M \to 1M \mid 1E$	$\{1^m 0^n \mid m > n\}$
$L \to L0 \mid E0$	$\{1^m 0^n \mid m < n\}$
$E \to \epsilon \mid 1E0$	$\{1^n 0^n \mid n \geq 0\}$

Explanation. The complement of L can be described as $\{1^m0^n \mid m \neq n\}$. Then consider the cases where m is More than n and m is Less than n. The case where m is Equal to n is used as a basis. The non-terminal L adds at least one 0's to the basis E so that $m < n$ always holds. The non-terminal M adds at least one 1's to the basis E so that $m > n$ always holds. Finally, S is produced from the union of M and L. The union of $m < n$ and $m > n$ produces $m \neq n$.

■