CS/ECE 374 FALL 2018
Homework 5 Problem 2

Zhe Zhang (zzhan157@illinois.edu)
Ray Ying (xinruiy2@illinois.edu)
Anqi Yao (anqiyao2@illinois.edu)

2. Let Σ be a finite alphabet and let L_1 and L_2 be two languages over Σ. Assume you have access to two routines IsStringIn$L_1(u)$ and IsStringIn$L_2(u)$. The former routine decides whether a given string u is in L_1 and the latter whether u is in L_2. Using these routines as black boxes describe an efficient algorithm that given an arbitrary string $w \in \Sigma^*$ decides whether $w \in (L_1 \cup L_2)^*$. To evaluate the running time of your solution you can assume that calls to IsStringIn$L_1(u)$ and IsStringIn$L_2(u)$ take constant time. Note that you are not assuming any property of L_1 or L_2 other than being able to test membership in those languages.

Solution: boolean values belong[1],belong[2],...,belong[n] are used to indicate whether the substring $w[i,...,n] \in (L_1 \cup L_2)^*$. The outer loop goes from the end of the string w to the start of the string w, and i indicates the index. The inner loop goes from i^{th} position to the end of the string w, and j also indicates the index. $j = n$ is a special case. When $j \neq n$ and belong[j+1] = TRUE which means substring $w[j+1,...,n] \in (L_1 \cup L_2)^*$ so that we only need to check whether $w[i,...,j] \in L_1$ || $w[i,...,j] \in L_2$ or not. If it is in either L_1 or L_2, we set belong[i] = TRUE which means the substring $w[i,...,n] \in (L_1 \cup L_2)^*$. At the end, if belong[1] = TRUE, then $w \in (L_1 \cup L_2)^*$

```
FUNCTION(w):
  boolean belong[1,2,...,n]
  for i ← n down to 1
    belong[i] = FALSE
    for j ← i to n
      if (j==n)
        if (IsStringInL1(w[i,...,n]) || IsStringInL2(w[i,...,n]))
            belong[i] = TRUE
            break
      else
        if (belong[j+1] && (IsStringInL1(w[i,...,j]) || IsStringInL2(w[i,...,j])) )
            belong[i] = TRUE
            break
  return belong[1]
```

The resulting algorithm runs in $O(n^2)$ time. ■