**CMPS 130**

**Homework 4**

**Patrick Tantalo**

**Seongwoo Choi**

**Chapter 2**

**Problem 1k**

The Finite automata accept the string that contain substring aba and the substring bab, which means that the string should contain the sub strings aba and bab. The finite automaton that accepts the string that contains the sub strings aba and bab is this:

**Problem 14**

Let x1 and x2 be distinct prefixes of x, and suppose x1y1 = x2y2 = z and |x1| < |x2|. Then x1y1 is not the element of L and x2y2 is the element of L.

**Problem 17**

1. L-distinguishable: Two distinct strings x and y are said to be L-distinguishable, if there exists another string z such that any one of xz or yz belongs to L. The set of alphabets {a, b}\* contains {rambda, a, b, aa, ab, ba, bb, aaa, aab, …….}. Assume that x = bba and y = bbba. Let z be another string and assume that z = b. Concatenation of xz is bbab. The string is not acceptable by the language L. Concatenation of yz is bbbab. The string is not accepted by the language L. As, xz is not an element of L and yz is not an element of L. So, the strings x and y are indistinguishable. Therefore, the strings x and y are not L-distinguishable.
2. Pairwise distinguishable: A set of strings S is said to be pairwise distinguishable with respect to language L, only if every pair of strings in S are distinguishable with respect to L. The infinite set consisting of pairwise distinguishable elements are S = {a^n | n ≥ 0}. For example, let the two distinct strings defined by set S is x = a and y = aa. Assume another string z belonging to alphabet set {a, b} such that z = b. Concatenation of xz is ab, which is a member of L when n = 1. Concatenation of yz is aab, which is not a member of L when n = 1. Concatenation of yz is aab, which is not a member of L. So, x and y are distinguishable strings in the set S. Therefore, the set S consists of pairwise distinguishable elements.

**Problem 21a**

Assume in each part that 0 ≤ i < j. The string ba 2i distinguishes ai and aj.

**Problem 26**

According to the pumping lemma, if n is the number of states in an FA accepting L, and there is a string x in L with |x| ≥ n, then L must be infinite.

**Problem 33**

There are |x| + 2 equivalence classes, one for each prefix of x and one containing all the non-prefixes

**Problem 35**

Suppose xILΛ and x≠Λ. then for every I ≥ 1, either both the strings xiy and xxiy are in L or neither is, which means that xiyIL i+l y. Since [Λ] contains all the strings xi it must be infinite.

**Problem 40**

The part a and b. Consider that there is a string x that is an element of {a,b}\*, and let k = na(x) – nb(x). Saying that k = 0 is the same as saying that x ∈ L. If k > 0, then x has an excess of k a’s, so that for any z, xz ∈L iff z has an excess of k b’s. Similar to how we got this value, if k<0, xz ∈L iff z has an excess of –k a’s. This means that if na(x) – nb(x) = na(y)-nb(y), then x and y have the property that xz and yz will be in L for precisely the same strings z-i.e., x and y are equivalent. Conversely, if na(x) – nb(x) ≠ na(y) – nb(y), then any string z for which nb(z) – na(z) = na(x) – nb(x) distinguishes x and y.

For part c, we know that parts a and b are saying that an equivalene class containing a string x is determined by the difference na(x) – nb(x): another string y will be in this equivalence class precisely if na(y) – nb(y) is the same value as for x. This means that for every integer k, there is an equivalence class containing all those strings x for which the difference is k. Another way to say this is to say that the equivalence classes are …, [bbb], [bb], [b], [Λ] = L, [a], [aa], [aaa],…

For each k > 0, [bK] = {x | nb(x) - na(x) = k}, and [aK] ={x | na(x) – nb(x) = k},