**COT 4210 Fall 2014 Sample Problems**

1. Draw a DFA to recognize the set of strings over **{a,b}\***that contain the same number of occurrences of the substring **ab** as of the substring **ba**.

**2**. Present the transition diagram or table for a DFA that accepts the regular set denoted by the expression **(0+1)\* (010 + 11) (0 + 1)\***

**3.** Consider the following assertion:

Let **R** be a regular language, then any set **S**, such that **S ∪ R = S**, is also regular.

State whether you believe this statement to be True or False by circling your answer.

**TRUE** **FALSE**

If you believe that this assertion is True, present a convincing argument (not formal proof) to back up your conjecture. If you believe that it is False, present a counterexample using known regular and non-regular languages, **R** and **S**, respectively.

**4.** Assume that **L1** and **L1∩L2** are both regular languages. Is **L2** necessarily a regular language? If so, prove this, otherwise show that **L2** could either be regular or non-regular.

**5.** Let **L** be defined as the language accepted by the finite state automaton **A**:

**A**

**B**

**C**

**D**

**E**

1

λ

0

1

0,1

λ

0

1

**A:**

**a.)** Fill in the following table, showing the -closures for each of **A**’s states.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **State** | **A** | **B** | **C** | **D** | **E** |
| **-closure** | **{ }** | **{ }** | **{ }** | **{ }** | **{ }** |

**b.)** Convert **A** to an equivalent deterministic finite state automaton. Use states like **AC** to denote the subset of states **{A,C}**. Be careful -- -closures are important.

**6.** Let **L** be defined as the language accepted by the finite state automaton **A**:

**A**

**B**

**C**

**A:**

0,1

0

0

1

0,1

0

Change **A** to a GNFA (although I don’t see the need for all those φ transitions). Now, using the technique of replacing transition letters by regular expressions and then ripping (collapsing) states, develop the regular expression associated with **A** that represents the set (language) **L**.

**7.** Let **L** be defined as the language accepted by the finite state automaton **A**:

**A**

**B**

**C**

**A:**

0,1

0

0

1

0,1

0

Using the technique of regular equations, develop the regular expression associated with **A** that represents the set (language) **L**.

**8.** Given a finite state automaton denoted by the transition table shown below, and assuming that **5** and **6** are final states, fill in the equivalent states matrix I have provided. Use this to create an equivalent, minimal state automaton. The start state is **1**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **a** | **b** | **c** |  |  |  |  |  |  |  |
| **1** | **4** | **6** | **2** |  | **2** |  |  |  |  |  |
| **2** | **5** | **2** | **1** |  | **3** |  |  |  |  |  |
| **3** | **4** | **5** | **4** |  | **4** |  |  |  |  |  |
| **4** | **5** | **4** | **3** |  | **5** |  |  |  |  |  |
| **5** | **4** | **6** | **1** |  | **6** |  |  |  |  |  |
| **6** | **2** | **6** | **3** |  |  | **1** | **2** | **3** | **4** | **5** |

**Don’t forget to construct and write down your new, equivalent automaton!!**

**9.** Use the Pumping Lemma to show that the following languages are not regular:

**a.**) **L = { an bm ct | n > m** or **n > t,** and **n, m, t** ≥ **0 }**

**b.) L = { an bm | n ≤ m,** and **n, m** ≥ **0 }**

**c.** Let **NonPrime** = { **aq** | **q is not a prime** }

This language is not easily shown non-regular using the Pumping Lemma. However, we already saw in class how to prove that **Prime** = { **ap** | **p is not a prime** } is non-regular using the Pumping Lemma. Explain how you could use this latter fact to show **NonPrime** is non-regular.

**10.** DFAs accept the class of regular languages. Regular expressions denote the class of regular sets. The equivalence of these is seen by a proof that every regular set is a regular language and vice versa. The first part of this, that every regular set is a regular language, can be done by first showing that the basis regular sets (**Ø , {  } , { a | a ∈ ∑ }**) are each accepted by a DFA.

**i.)** Demonstrate a DFA for each of the basis regular sets.

**Ø**

**{  }**

**{ a }**

Let **L1** be generated by the DFA **A1 = ( Q1 , ∑ , δ1 , q1 , F1 )** and **L2** be generated by the DFA   
**A2 = ( Q2 , ∑ , δ2 , q2 , F2 )**.

**ii.)** Present a construction from **A1 , A2** that produces a DFA **A3** for **L1** ∪ **L2**.   
Hint: Cross product.

**iii.)** What remains to be done to show that every regular set is a regular language? Don’t do the proof; just state what two steps still need to be done.