**COT 4210 Fall 2014 Sample Problems with Solutions**

**1.** 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** | **{ A }** | **{ B , C }** | **{ C }** | **{ D, E }** | **{ E }** |

**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.

**A**

**BC**

**BCDE**

1

1

1

0

0

**A:**

**Λ**

0,1

0

**2.** 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 ripping (collapsing) states, replacing transition letters by regular expressions, develop the regular expression associated with **A** that generates **L**. I have included the diagrams associated with removing states **A**, **B**, then **C**, in that order.

**S**

**A**

**B**

**C**

**F**

**A:**

λ

0+1

0

0

1

0+1

λ

0

**S**

**B**

**C**

**F**

**A:**

0+1

0

1

0+1+0(0+1)

λ

00

0

**S**

**C**

**F**

**A:**

000\*1+0+1+0(0+1)

λ

0+1+00\*1

**S**

**F**

**A:**

(0+1+00\*1)(000\*1+0+1+0(0+1))\*

**3.** Let **L** be recognized by the DFA, **A=( Q , Σ , δ, qo , F )**, where **|Q|=N**.

Use the Pumping Lemma to show that the following language,   
**L = { an bm ct | n > m** or **n > t,** and **n, m, t** ≥ **0 }**, is not regular.

Proof by contradiction:

Assume **L** is regular and let **N** be the number from the P.L. Clearly

**aNbN-1cN-1 ∈ L**

By P.L., **aNbN-1cN-1 ≡ uvw**, where **|uv| ≤ N** and **uw ∈ L**, since we can pump as **uv0w**. But then, if we compose the expression as the following: **aN-|v| a|v| bN-1cN-1**, when we remove **|v| a**’s, via pumping, and we end up with **aN-|v|bN-1cN-1** belonging to **L** Since, **|v| > 0**, the number of **a**’s is less than or equal to the number of **b**’s and **c**’s, which implies **aN-|v|bN-1cN-1 ∉ L**, which is a contradiction of our assumption, and therefore **L** is not regular.

**4.** Analyze the following language, **L**, proving it non-regular by showing that there are an infinite number of equivalence classes formed by the relation **RL** defined by:   
 **x RL y** if and only if [**z** ∈ **{a,b,c}\***, **xz** ∈ **L** exactly when **yz** ∈ **L ]**.   
where **L = { an bm ct | n** > **m** > **t }**.   
You don’t have to present all equivalence classes, but you must demonstrate a pattern that gives rise to an infinite number of classes, along with evidence that these classes are distinct from one another.

Clearly, **aibi-1ci-2 ∈ L**, **ai+1bi-1ci-2** and also **ai+1bici-1 ∈ L** but **aibici-1 ∉ L**, which implies, **ai RL aj** iff **i = j**. Since both **ai** and **aj** are **RL** distinguishable when **i ≠ j**, then there are an infinite number of equivalence classes. Thus **L** is non-regular.

**4.** Consider the regular grammar **G**:

**S → 0 S | 1 A**

**A → 0 S | 0 A | 1 B | **

**B → 1 S | 0 B**

**a.)** Present an automaton **A** that accepts the language generated by the **G**:

**S**

**B**

**A:**

0

0

0

1

0

1

1

**A**

**b)** Regular grammars generate 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 ∈ Σ }**) are each generated by a regular grammar over the alphabet  **∑**.

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

**Ø G = { {S}, Σ, S, Ø }**

**{ } G = { {S}, Σ, S, {S → λ } }**

**{ a } G = { {S}. Σ, S, {S → a} }**

Let **L1** be generated by the regular grammar **G1 = ( N1 , Σ , S1 , P1 )** and **L2** be generated by the regular grammar **G2 = ( N2 , Σ , S2 , P2 )**, where **N1 ∩ N2 = Ø**.

**ii.)** Present a construction that produces a regular grammar for **L1** • **L2**.

**G = { N1 ∪ N2, Σ, S1, P }**

**P** = **{ X → wS2 | ∀** rules in **P1** of the form **X → w**, where **X ∈ N1­** and **w ∈ Σ** } ∪

**{ X → wY | ∀** rules in **P1** of the form **X→wY**, where **X, Y ∈ N1** and **w ∈ Σ** } ∪ **P2**

Why is the property **N1 ∩ N2 = Ø** needed here?

To prevent rules from the different grammars from mixing with one another when generating the new transition set.

**iii.)** What remains to be done to show that every regular set is a regular language? Don’t do the proof, just state what needs to be done.

Prove closure under union and Kleene\*.

**5.** Present a Mealy Model finite state machine that reads an input **x ∈ {0, 1}\*** and produces the binary number that represents the result of subtracting **10** from **x** (assumes all numbers are positive, including results). Assume that **x** is read starting with its least significant digit.  
Examples: **0010 → 0000; 1000 → 0110; 0001 → 1111** (wrong answer due to going negative)

**-0**

**-1**

0/0,

1/1

0/1

1/0

0/0, 1/1

**-10**