**Assignment#1; Due January 22 at start of class  
Review of Formal Languages**

Consider some language **L**. For each of the following cases, write in one of **(i)** through **(vi)**, to indicate what you can say conclusively about **L**’s complexity, where

**(i)** **L** is definitely regular

**(ii)** **L** is context-free, possibly not regular, but then again it might be regular

**(iii)** **L** is context-free, and definitely not regular

**(iv)** **L** might not even be context-free, but then again it might even be regular

**(v)** **L** is definitely not regular, and it may or may not be context-free

**(vi) L** is definitely not even context-free

Follow each answer with example languages **A** (and **B**, where appropriate) to back up the complexity claims inherent in your answer; and/or state some known closure property that reflects a bound on the complexity of **L**.

**Example.) L = A ∪ B**, where **A** and **B** are both context free, and definitely not regular

**L** can be characterized by **Property (ii)**, above.

**L** is context-free, since the class of context-free languages is closed under union.

**L** can be regular. For example,

**A = { an bm | m ≥ n }**, **B = { an bm | m ≤ n }**,

**L = A ∪ B = { an bm | n, m ≥ 0 }**, which is regular since it can be represented by the regular expression **a\*b\***.

But **L** is in general not guaranteed to be regular, e.g., if we just make **A** and **B** the same context-free, non-regular set, then **L = A ∪ A = A**, which is not regular.

**a.) L = A ∩ B**, where **A** and **B** are both context-free, non-regular

**b.) L = A ∩ B**, where **A** is context-free, non-regular and **B** is regular

**c.) A = B – L**, where **A** is context-free, non-regular and **B** is regular

**d.) A ⊂ L**, where **A** is Regular