COSC 362 Test #3 – Wed. 5/3 in class, plus take-home part due Mon. 5/8

Here are some types of problems to expect to see, and/or information you'll be expected to know, for Test #3. This is not necessarily an exhaustive list, but most (if not all) of the test will be based on the list shown below. Test questions will, whenever possible, be similar to problems assigned as recommended or collected homework.

Items marked with an asterisk (*) below would appear only on the take-home part of the test, if they are included at all. Items without an asterisk may appear on either part of the test.

"Be aware of…", "Know…", or "be familiar with…" items indicate topics which may be included on the in-class part of the test in the context of true/false, multiple choice, or short answer questions.

- Remove a λ, unit, or useless production from a given grammar
- Convert a given grammar to Chomsky normal form
- Convert a given grammar to Greibach normal form
- * Use the CYK algorithm to determine whether a given string is in the language generated by a given grammar

- * Given a language, design a npda that accepts the language. (You may do this by writing out a set of transition rules or by drawing a transition graph.)
- * Given an npda, determine whether a given string is accepted by the npda, and/or determine the language that is accepted by the npda
- * Given a context-free grammar, G, design a npda, M, such that $L(M) = L(G)$.
- Be aware of the more general result (not fully explored in class) that a language is context free <u>if and only if</u> it can be accepted by some npda. (This is the main point of Section 7.2.)

- Be familiar with the contents of the Chapter 8 summary handout from the class web page. In particular, you'll be expected to… (possibly without looking at the handout)
    - Know some examples of context-free languages that are not regular
    - Know some examples of languages that are not context-free
    - Know the operations under which the family of context-free languages is closed (union, concatenation, star-closure), as well as operations under which this family is not closed (intersection, complementation)
    - Be able to show (via verbal explanation, and/or by demonstrating with a given example) that the union or concatenation of two context-free languages is context-free

- * Given a language, design a Turing machine that accepts it. (You may do this by writing out a list of transition rules or by drawing a transition graph.)
- * Given a Turing machine, determine whether a given string is accepted, and/or determine the language that is accepted by the Turing machine
- * For the "combining Turing machines" supplements handout from the class web page...
    - * Given a "block diagram" (that is, a diagram of the type shown in that handout which utilizes one or more "building block" machines), be able to interpret what algorithm the diagram is describing
    - * Given an algorithm (i.e., a Turing machine), create a block diagram to carry out the algorithm (similarly to the examples in the handout)

- Be familiar with the statement of "Turing's thesis," as well as why it is significant to the theory of computation. Related to this (see Ch. 10), be familiar with what is meant by the "computing power" of a class of automata, and by the general conclusion that neither minor variations (section 10.1) nor major enhancements (section 10.2/10.3) of Turing machines are any more "powerful" than standard Turing machines.

- * Be familiar with the concept of a "Universal Turing machine," as well as some of its consequences
  - * Given a Turing machine and/or an input string, be able to give their unary "encoding" as defined in Section 10.4
  - * Conversely, be able to "decode" a given string whose unary encoding is given
- For infinite sets, know what is meant by "countable," as well as a few familiar examples of countably infinite sets (e.g. the natural numbers, the integers). In particular, be able to explain the difference between "countable" and "uncountable" for infinite sets.
- Know that the set of all Turing machines is countable, but the set of all languages on a given alphabet is uncountable. (You won't be asked to prove this last part, but it's an important result going forward, so make sure you're aware of it!)

In-class (50 minutes) vs. take-home (hopefully 1 hour, but schedule yourself 2 hours to be safe): Everything not marked with an asterisk is fair game for the in-class part of the text, with the restriction that any problem on this part will be one that *can* be done quickly (if approached in the correct way), and does not (or, at least, *should* not!) require access to the textbook.