Homework 3, CS301, McConnell, Spring '09

Due Thursday 2/26 at the beginning of class.

Modified 2/23 6:41 to correct the book's mistake on problem 2.16(i).

Modified 2/24 18:00 to remove a mistake in a hint on 2.16.

- Exercise 2.14
- Exercise 2.15
- Exercise 2.16, except for part a, which I give a solution for.

  Each problem has many solutions. To make it feasible to grade them, I will ask you to come up with a *shortest* regular expression for each language. The length of the regular expression is the number of characters needed to type it out, counting the union operator, the Kleene-star operator, and parentheses. You don't need to count the concatenation operator, which doesn't require an explicit character in the expression.

  For example, two expressions for the language of all strings on $\Sigma = \{a, b\}$ are $(a \cup b)^*$ and $(a^*b^*)^*$. The first requires only six characters and the latter requires seven. You should use the first, not the second.

  To get you started, I will give you the answer to part a: b^*ab^*. *Hints: For part c:* $(b \cup a)^*a(b \cup a)^*a(b \cup a)^*$ *is not the shortest. If there are two a's, there must be two a's that precede all others.*

  Matt Dunlap has pointed out that the the description of the language of part i given by the book in parentheses is different from the original. For example, the string "a" belongs in the language given in parentheses, but not the original. Ignore the description in parentheses. **Thanks to Jon Roelof and Conner Petzold for pointing out an error in one of my own hints, which I have removed.**
- Exercise 2.17: We want two different proofs, a regular expression for the language and a DFA that recognizes the language. Since there is more than one solution, you must find a shortest possible regular expression, and a DFA with five states.
- Exercise 2.19a. The point we're looking for is that you understand the algorithm that you would program it into the computer. Don't take common-sense shortcuts that aren't part of the algorithm as it's described in the book. If the algorithm says to insert an edge labeled with the empty string, make sure this appears in your diagram.
- Exercise 2.22, parts a and d. Read about Algorithm $\mathcal{B}$ and understand why it works before you begin. We went over the example of Figure 2.40, which required us to apply and justify the correctness of each of the three rules the algorithm uses for eliminating a state. However, there are some additional issues that come up in the general case that are explained on pages 38 and 39; be sure to read about these.

  Once again, the point is that you understand that the procedure is a formal algorithm. Don't take common-sense shortcuts. If it says to take the union of two edge labels, make sure you do this without simplifying the expression. Show the picture of the

expression diagram after each state is eliminated. Eliminate them in ascending order of state number.