CS 3240 Homework

Spring 2010

Chapter 2

Section 2.1

Do exercises 2abd, 5b, 7b, and 9c. (Give transition graphs). Also do the following exercises.

A. Referring to the slide with the finite automaton for an automatic door (see Chapter 1 slides), construct an automaton that would model an automatic that swings *both ways*, *inward* and *outward*. Keep in mind that the sensor pads are close enough to the door that if the door moves, it will hit the person standing in the way. Give both a transition graph and a transition table. Don't let any customer get hit!

B. Construct a DFA that accepts bit strings representing numbers ≡ **2 mod 4** or **3 mod 4**.

Section 2.2

Do exercises 7, 12, and 21, and the following exercise.

A. Write an NFA (*not* a DFA!) that accepts strings over $\Sigma = \{a, b\}$ whose next-to-the-last letter is a b.

Section 2.3

Do exercises 1, 3, and 12, and the following exercise. Note: for #1, the alphabet is $\{a\}$. Note #2: the procedure on 59 is equivalent to the "combination table" technique I showed you (where you combine multiple states into a single, composite state, as needed). You can ignore page 59 if you like (it's a little obtuse for the non-mathematician :-). Note #3: L^R is the language where all the strings of L are reversed (see pages 17 and 19).

A. Convert the NFA in #A in Section 2.2 above to a DFA.

Section 2.4

Do exercise 1.

Appendix A

A. Draw a transducer (Mealy machine) that reads a bit string representing an unsigned integer and prints out its octal equivalent. You may assume that the string length is a multiple of 3.

B. Draw a transducer (Mealy Machine) that reads a pair of equal-length bit strings representing unsigned integers as input and prints out the largest of the two numbers.

(*For both of these problems the numbers may have leading zeroes*)

Chapter 3

Section 3.1

Do exercises 1, 4, 5, 16a, and 16b.

Section 3.2

Do exercises 1, 4d, 6, 10c, and 15. For 15, allow for signs, and for things like *0., .0, 1., etc.*

Section 3.3

Do exercises 1, 2, and 3.

Chapter 4

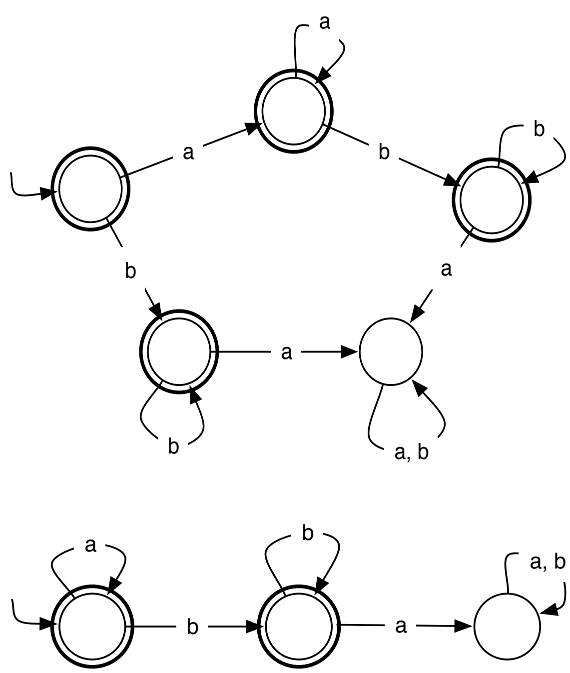
Section 4.1

Do exercises 2b, 6, and 8. On 2b, you can use the "combination table" method from slide 9. It's the same as the procedure in Theorem 4.1.

Section 4.2

Do exercises 6, 8, and 14, and the following:

A. Determine whether the following FAs represent the same language.



Section 4.3

Do exercises 3, 4b, and 4d.

Chapter 5

Section 5.1

Do exercises 2, 3, 5, 7c, 8b, 8c, 13b and 21.

Section 5.2-5.3

Do exercises 8, 12, and 13 in section 5.2 and the following:

A. Add exponentiation (^), modulus (%) to the grammar on slide 27 for Chapter 5. Give % higher precedence than * and ^ higher precedence than %. Note that ^ is right-to-left and % is left-to-right associative.

Chapter 6

Section 6.1:

Do exercises 6, 7, and 9. Simplify each answer as much as possible.

Section 6.2:

Do exercises 3, 4, 7, and 13. On 7, we want the graph of the *original* grammar.

Section 6.3:

Do exercise 1. Use the tabular method.

Chapter 7

Section 7.1:

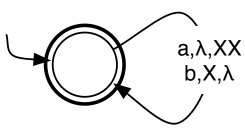
Do exercises 4c-e,i, 8, and the following problem:

A. Draw a NPDA that accepts the language over $\{a, b\}$ where the number of a's is exactly one greater than the number of b's.

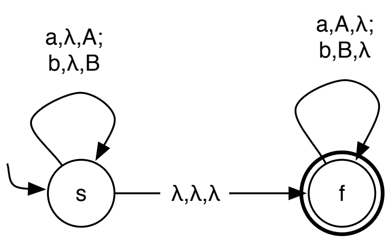
Section 7.2:

Do exercise 5 and the following problems.

A. Use the simple, 2-state "push and pop S" technique to find a CFG for the language accepted by the following PDA.



B. Convert the following PDA to a CFG by the generic process explained in class and on the web site. Verify by deriving *abbaabba* and *aaabbaaa*.



Section 7.3:

Do exercises 2 and 6. (*Hint:* Use a stack start symbol for these problems.)

Chapter 8

Section 8.1:

A. Is the language $a^nb^{3n}a^n$ context free? If it is, find a grammar for it. If not, use the pumping lemma to prove it.

Section 8.2:

Do exercises 8, 18, 23, and the following:

A. Find a CFG for the language over $\Sigma = \{a, b\}$ of strings that either start with an a or are of the form a^nb^n.
B. Find a CFG for the language over $\Sigma = \{a, b\}$ of strings whose prefix has an equal number of a's and b's and whose suffix is of the form a^nb^n.
C. Find a CFG for the language $(a^nb^n)^*$, $n > 0$.

Chapter 9

Section 9.1:

Do exercises 3, 7c, 7e, 16, and the following problems:

A. Construct a TM that computes x-y, where x and y are represented in unary notation. If y is greater than x, give 0 as the answer. The only symbols left on the tape should be a number of 1's, if $x > y$, or the symbol 0 if $x \leq y$.
B. Construct a TM that accepts the language $n_a(w) = n_b(w) = n_c(w)$ for all strings w over the alphabet $\sum = \{a,b,c\}$.

Chapter 10

Section 10.4:

Do exercises 1 and 2, plus the following problem.

A. Following the example in the slide set, construct a *queue machine* that accepts the language $n_a(w) = n_b(w) = n_c(w)$ for all strings w over the alphabet $\sum = \{a,b,c\}$.

Chapter 11

Section 11.1:

Do the second part of exercise 3 (showing an enumeration strategy for L^+, where L is an finite language.).

Section 11.2:

A. Using slide 32 as a guide, find an unrestricted grammar for the language $n_a(w) = n_b(w) = n_c(w)$. Use it to derive *abaccb*.

Chapter 12

Section 12.1:

Do exercise 5. (*Hint*: Given M and w, you need to create a machine that runs M on w when it receives w as input. What will you do for all other input?)