MA/CSSE 474 Homework #9 72 points

**General hint:** If it is very difficult to find a way to make the pumping theorem work for a given language that appears to be non-regular, consider the possibility that the language might actually be regular.

1. 8.8
2. (t-3-6) 8.8 de
3. (t-9) 8.9 This is a difficult problem. Begin thinking about it a few days before it is due.

Definitions of *maxstring* and *mix:* Examples 8.22 and 8.23.

4. (t-9) 8.10-a I.e., given a DFSM M = (K, Σ, δ, s, A) such that L(M)=L, construct a DFSM M*=(*K**, Σ, δ *, *s**, *A**) such that L(M*)=maxstring(L).
5. (t-6) 8.16a (this one is a little bit "logically tricky")
6. 8.16b
7. 8.21 (I like to put questions like these on exams)
8. (t-12) 8.21n (this is a nontrivial problem)
9. (t-3) 8.21o
10. 9.1 You can assume (and use without giving the details of the algorithms) any algorithms and decision procedures from chapter 9 or previous chapters.
11. (t-6) 9.1b See note below.
12. (t-6) 9.1d
13. (t-6) 9.1g. See note below.
14. (t-6) 9.1i

**9.1g:**
There is a small error in the statement of the problem. a* should be {a}*

**9.1b:**
Note that |L(M)| means "the number of elements in the language accepted by the machine M. Note that for some machines M, the language is countably infinite.

## Previous questions and answers from Piazza:

### Instructor announcement:

## HW9 alert!

HW9 is longer than any previous assignments, and it has a few problems that are more challenging than a typical assignment.Like HW2, and HW5, I suggest that you read the problems soon and begin thinking about them, even before you finish A8. I mentioned last week that the course would soon get significantly harder. A9 is by far the hardest assignment from the first 2/3 of the course.

"It would have been a waste of time to begin thinking about these problems 6 days before they were due!" said no 474 students on the due date.

**General question:** Is every Nonregular language countable in size? **A** Every language is countable (I.e. it is finite or countably infinite, because Σ is finite).

**General question:** Can we assume we know how to check if there are loops in a DFSM as the books assumes? **A** Yes. Unless a problem specifically states otherwise, for decision procedure problems, you may assume anything that is in an result in the book, the homework, or a class example or exercise.

**Question on #8: I'm stuck on how to prove this one. Any particular rule I should be using,,,?**

**Hint 1:** It's true. All of the credit is for showing that.

**Hint 2:** I am not sure how you could show this except by construction. I.e., describe how, given any non-regular language L, we can construct an infinite set T of regular languages, such that the intersection of all of the languages in T is exactly the set L.

**Hint 3:** The notion of "complement of a language" can be useful here.

**Student question:**
One thing that confuses me with this problem is that the hint suggests that there is a way to do describe every nonregular language as the intersections of infinite regular languages. However, I remember that one of our homeworks had us show how to create an intersection machine, meaning that the regular languages must be closed on intersection (because regular intersect regular can be written as FSM). How does doing intersection an infinite number of times allow for nonregular languages if the regular languages are closed under that operation?
**Student answer (endorsed by instructor): The** regular languages are closed under finite intersection. They are not closed under infinite intersection. This is similar to how finite sets are closed under union, but not infinite union.

## Instructor note: HW9: How NOT to write a decision algorithm!

A student began an "algorithm" for problem 12 (9.1d) with:

Let L = L(α) ∪ L(β).

A statement like this is fine for a proof, but not for an algorithm. What kind of data structure would hold L? Unless L is finite, it can't be a set of all of the elements.
The only ways we have to represent a language in general are a FSM or a regular expression.
So most of the steps in decision algorithms should generally begin "Build a FSM that... " or "build a regular expression that ..."
Once you have a FSM, you can count its states, or ask questions like those in section 9.1 of the textbook.

## Book example of Maxstring, last row in table

I'm currently looking over the table for maxstring in Example 8.22 of the textbook. In it, I can't quite tell what the superscript above the b is for the last row in the maxstring(L) column. Is it a 1? (And if so, is the idea that since "a" is in the language (if a* and b* were both epsilon "cases" (not my best wording, but I think I'm getting my idea across), we need to have the single instance of b to break up any possible rightward expansion?)
**Instructor answer:** The superscript is +, indicating one or more b's
For a little bit more on *maxstring*, see the Functions Over Languages video that is linked from Day 4 of the schedule page.

## 8.8

When this problem asks for "give a precise definition of the following languages:" is a set or regular expression sufficient? Or is it asking for something else?

**Instructor answer:** A regular expression or set notation is fine.

## Construction vs. proof of existence

**Instructor note:**
In HW 9, problem 8, you were supposed to show that for every non-regular language L, there exists an infinite set S of regular languages over L's alphabet such that L is the intersection of the elements of S.
Some students expressed their proofs as a construction, with a step that is something like:

$T = \{\ \}$

for each string w in the complement of L:

    $T = T \cup \{\Sigma^* - \{w\}\}$

The problem is that we can't do this, because it requires infinitely many steps.

A construction must be an algorithm.  It must terminate after a finite number of steps.

How is the above different than the exam problem where you did the construction for the reverse of a language, which might be infinite?  In that construction, we build a finite description of a NDFSM from a finite description of a DFSM; we never deal with the individual elements of the languages themselves.