

LING 106: Homework 5

Due: Monday, March 30, 2009, 5:00pm

Read the following carefully.

This homework is divided into two parts. Part One consists of problems 1-3. Everyone must do those.

Part Two consists of the problems 4-6. You must do those *if and only if* you scored below a B on the midterm. They consist of problems that cover material from the midterm; people who received a B or higher have already demonstrated their command of this material.

Part Two will be scored separately and factored into the final grades in a perfectly fair manner to be determined.

Finally, in case it's not obvious by now: if something isn't clear, *ask*.

1. CONSTRUCTING NFAS

For each of the following languages, $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$, construct the simplest NFA. Some guidelines:

- Having fewer states is simpler than having more;
- Having fewer final states is simpler than having more;
- Having fewer transition arrows is simpler than having more;
- A transition with ϵ is simpler than a transition with a symbol of the alphabet

Don't spend too much time stressing over whether four states, of which one is a final state, is simpler than three states, of which two are final states. These are guidelines. You'll get marked down a little for egregious complexity, but little variations are fine.

- a. The language $(\mathbf{ab})^*(\mathbf{bc})^*$
- b. The language $\{\epsilon, \mathbf{a}, \mathbf{ba}, \mathbf{cab}, \mathbf{aab}, \mathbf{ccc}\}$
- c. The language $\{s \mid s \text{ contains neither three } \mathbf{as} \text{ in a row nor three } \mathbf{bs} \text{ in a row}\}$

2. TWO PROOFS

I stated in class that the set of regular languages is closed under the operations union, intersection, concatenation, and star. We gave a proof by construction for the first two operations: given two arbitrary DFAs, we gave a method for finding a machine that models the union of the languages they model, and similarly for the intersection. The concatenation and star theorems are harder to prove with DFAs, but now that we know that NFAs also model regular languages...

- a. Prove that regular languages are closed under concatenation. That is, given two arbitrary FSAs that model languages L and M , show how to construct a FSA that models the concatenation of L and M . (Remember that $L \circ M = \{l \circ m \mid l \in L \text{ and } m \in M\}$.)
- b. Prove that regular languages are closed under the star operator. That is, given an arbitrary FSA that models language P , show how to construct a FSA that models P^* . (Remember that $P^* = \{x_0 \circ x_1 \circ \dots \circ x_k \mid k \geq 0 \text{ and each } x_i \in P\}$.)

Note: these may not seem easy at first! The key to both of them is to think about what “concatenation” and “star” mean.

The ideal answer will have the same form as the proofs from class, i.e. one that specifies the 5-tuple for the concatenation machine from the 5-tuples of L and M , and similarly for the star machine. If this is overwhelming, tackle the task in stages:

- Try constructing the machine for a particular language or two.
- Describe the particular method for those machines.
- Describe the general method in plain English.
- Describe the general method formally.

3. THE PUMPING LEMMA

Remember that the Pumping Lemma guarantees that *there is some number p* , such that all strings of length p or greater will have a non-empty “pumpable substring” in the first p symbols.

- a. For the regular language $L = \{s \mid s \text{ contains at most three } \mathbf{1}s\}$ and $p = 4$, show that the following strings in L obey the pumping lemma: **10101**, **1110**, **111**.
- b. Given $\Sigma = \{\mathbf{a}, \mathbf{b}\}$ and $L = \{s \mid s \text{ contains at most fifteen } \mathbf{a}s \text{ and at most fifteen } \mathbf{b}s\}$, explain why L obeys the pumping lemma.
- c. Prove that $L = \mathbf{xy}^n\mathbf{xy}^n\mathbf{y}^n$, where $n \geq 0$, is not regular.

4. SET THEORY REVISITED

Given the set of strings B :

$$B = \{\mathbf{y}, \mathbf{zy}, \mathbf{yzy}, \mathbf{zyzy}, \mathbf{yzyzy}, \mathbf{zyzyzy}, \mathbf{yzyzyzy}, \mathbf{zyzyzyzy}, \dots\}$$

PART ONE: Express B in predicate notation.

That means that you should have something like $B = \{x \mid \dots\}$, where the ellipses are a description of the strings x that are members of B . Note that the following do *not* work:

- $\{x \mid x \text{ contains only } \mathbf{y} \text{ and } \mathbf{z}\}$. The predicate is true of everything in B , but it's also true when $x = \mathbf{yyy}$, when $x = \mathbf{zzyyz}$, when $x = \mathbf{z}$, and so forth.
- $\{x \mid B \text{ is a set containing } \dots\}$. The predicate should describe each string x that's in B , not B itself.

You want a predicate that describes all the members of B , and only the members of B . You should be able to say, "Is this string a string such that it [...] ? If so, it belongs in the set; if not, it doesn't." For instance:

- "Is this string a string such that it contains only \mathbf{y} and \mathbf{z} ?" If your string is \mathbf{yzy} , the answer is "yes", so it belongs in the set; but if your string is \mathbf{yyy} , the answer is also "yes", even though \mathbf{yyy} is not in set B ; that's why this predicate isn't correct.

PART TWO: Express B in recursive notation.

Remember, to be in recursive notation, you need:

- A finite number of elements that you assert to be part of B . There can be more than one.
- A set of rules telling you how to get from any arbitrary element of B to another element of B . (e.g., "if x is in B , then this other thing (expressed in terms of x) is in B ") There can be more than one.
- The assertion that nothing else is in B .

Also remember that, even for set $C = \{x \mid x \text{ contains only } \mathbf{y} \text{ and } \mathbf{z}\}$, the following is not part of a legitimate recursive notation:

- If $x \in C$, then x contains only \mathbf{y} and \mathbf{z} .

That's a predicate; but it doesn't give you a *method* for turning one element of C into a new element of C . You want a method, i.e. a procedure. If in doubt, try taking your answer and performing the same series of steps you used in the first question of Part One.

5. FUNCTIONS

For each of the following relations, explain whether the relation is a function or not. If the relation is a function, explain whether it is many-to-one or one-to-one, and explain whether it is into or onto. (Explain, not just state!)

Tip: there are four different concepts at work here. They are: whether everything in the domain is used at least once, whether everything in the domain is used only once, whether everything in the range is used at least once, and whether everything in the range is used only once. Be certain that you know how these four concepts relate to functionhood, onto-ness, and one-to-one-ness.

The relations use the following sets:

- \mathbb{N} = the set of “counting numbers”, $\{0, 1, 2, 3, 4, 5, \dots\}$
- L = the set of all people currently living
- P = the set of all people, living or dead

ONE: $f \subseteq \mathbb{N} \times \mathbb{N}$, where $f(\langle x, y \rangle)$ if $x = y$.

TWO: $g \subseteq \mathbb{N} \times \mathbb{N}$, where $g(\langle x, y \rangle)$ if $y = x^2$.

THREE: $h \subseteq \mathbb{N} \times \mathbb{N}$, where $h(\langle x, y \rangle)$ if $x = y^2$.

FOUR: $m \subseteq (\mathbb{N} \times \mathbb{N}) \times \mathbb{N}$, where $m(\langle \langle x, y \rangle, z \rangle)$ if $x + y = z$

FIVE: $\varphi \subseteq L \times P$, where $\varphi(\langle x, y \rangle)$ if y is the (biological) mother of x

SIX: $\alpha \subseteq \mathbb{N} \times L$, where $\alpha(\langle x, y \rangle)$ if x is the age of y , in years (rounded off)

SEVEN: $\beta \subseteq L \times \mathbb{N}$, where $\beta(\langle x, y \rangle)$ if y is the age of x , in years (rounded off)

Reminder: you can read “ $f(\langle x, y \rangle)$ if [condition]” as “ $\langle x, y \rangle$ is an element of relation f if and only if [condition]”—for instance, in the first question, “ $f(\langle x, y \rangle)$ if $x = y$ ” means “ $\langle x, y \rangle$ is an element of relation f if and only if $x = y$ ”. (Remember, functions are relations which are in turn just sets of tuples.)

Its often easier to think of functions in terms of inputs and outputs—such as “ $f(x) = y$ ” for number one. But this notation is used for functions, so using it would presume that these actually *are* functions to begin with. Part of the problem is sorting out relations that are functions from those that are not.

6. THE δ FUNCTION

For each of the following δ s, state whether δ can be the transition function for a deterministic finite state automaton. Explain your answer; if δ cannot, explain how you could modify it *as little as possible* to make it a possible transition function.

(Note: you won't be able to evaluate what strings the machine accepts without knowing the start state and accept states, so don't worry about those.)

ONE: $\delta(\langle q_0, \mathbf{0} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{1} \rangle) = q_0$
 $\delta(\langle q_0, \mathbf{1} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{0} \rangle) = \text{dead}$

TWO: $\delta(\mathbf{0}) = q_1$
 $\delta(\mathbf{1}) = q_0$
 $\delta(\mathbf{2}) = q_1$
 $\delta(\mathbf{3}) = q_0$

THREE: $\delta(\langle q_0, \mathbf{a} \rangle) = q_0$
 $\delta(\langle q_1, \mathbf{a} \rangle) = q_0$
 $\delta(\langle q_1, \mathbf{x} \rangle) = q_0$
 $\delta(\langle q_0, \mathbf{x} \rangle) = q_1$

FOUR: $\delta(\langle \text{start}, \mathbf{\#} \rangle) = \text{whoosh}$
 $\delta(\langle \text{start}, \mathbf{\$} \rangle) = \text{tweet}$
 $\delta(\langle \text{start}, \mathbf{\&} \rangle) = \text{start}$
 $\delta(\langle \text{whoosh}, \mathbf{\#} \rangle) = \text{whoosh}$
 $\delta(\langle \text{whoosh}, \mathbf{\$} \rangle) = \text{start}$
 $\delta(\langle \text{tweet}, \mathbf{\#} \rangle) = \text{tweet}$
 $\delta(\langle \text{tweet}, \mathbf{\&} \rangle) = \text{start}$

FIVE: $\delta(\langle q_0, \mathbf{t} \rangle) = q_0$
 $\delta(\langle q_0, \mathbf{h} \rangle) = q_1$
 $\delta(\langle q_0, \mathbf{e} \rangle) = q_2$
 $\delta(\langle q_1, \mathbf{t} \rangle) = q_2$
 $\delta(\langle q_1, \mathbf{h} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{e} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{t} \rangle) = q_1$
 $\delta(\langle q_2, \mathbf{h} \rangle) = q_0$
 $\delta(\langle q_2, \mathbf{e} \rangle) = q_2$